# IMPLEMENTING AND TESTING UPSTREAM CORNER BALANCE METHODS IN PDT

An Undergraduate Research Scholars Thesis

by

ROBERT J. SEAGER, JR.

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Marvin Adams

May 2014

Major: Nuclear Engineering

# TABLE OF CONTENTS

# ABSTRACT

Implementing and Testing Upstream Corner Balance Methods in PDT . (May 2014)

Robert J. Seager, Jr.
Department of Nuclear Engineering
Texas A&M University

Research Advisor: Dr. Marvin Adams
Department of Nuclear Engineering

Among the many applications that require solutions of the particle transport equation are nuclear reactors, medical imaging, medical therapy, photon interaction in the atmosphere and oceans, industrial systems, and radiative transfer problems arising from many areas including astrophysics and inertial confinement fusion. For most transport problems of practical interest it is not possible to find analytic solutions. Thus, discrete approximations must be employed. Many spatial discretization methods exist that can provide accurate solutions if the spatial mesh is sufficiently fine, but the computational cost of transport problems would often prohibitively large if such a fine mesh were to be employed. Obtaining an accurate solution on a coarser mesh is significant. Upstream corner balance (UCB) methods are a specific family of spatial discretization methods that have some advantages over other methods. This research assesses the strengths and weaknesses of UCB by implementing, testing, and comparing UCB methods against other spatial discretization methods in the particle transport code PDT. In the test problems simulated, UCB was found to outperform the piecewise linear discontinuous (PWLD) method and the corner balance-step (CB-STEP) method. Based on these encouraging results, this research lays the foundation for further testing and development of UCB methods on a larger scale. With continued research, a robust, accurate, and efficient spatial discretization method could result.

# ACKNOWLEDGMENTS

I would like to acknowledge Dr. Marvin Adams for his guidance and assistance throughout this project.

I would also like to acknowledge Mr. Daryl Hawkins and Mr. Michael Adams for their assistance with computer coding issues.

# CHAPTER I

# AN INTRODUCTION TO PARTICLE TRANSPORT SIMULATIONS

Among the many applications of computational methods in nuclear engineering is particle transport simulation as it applies to radiative transfer and neutronics problems. The process of particle transport can be modeled by a transport equation. Among the many applications that require solutions of the transport equation are nuclear reactors, medical imaging, medical therapy, photon interaction in the atmosphere and oceans, industrial systems, and radiative transfer problems arising from many areas including astrophysics and inertial confinement fusion. However, for most transport problems of practical interest it is not possible to find analytic solutions. Thus, discrete approximations must be employed. The method by which the spatial variable is discretized is one of the most important factors that determine how accurately a transport problem is solved. Once a problems volume has been divided into "cells," a combination of approximate and exact equations are used to relate unknown quantities to known quantities. In a given cell, for particles of a given energy moving in a given direction, the known quantities are rates at which particles are "born" from sources and rates at which they enter the cell through each "incoming" surface. Unknown quantities are rates at which particles have collisions within the cell and rates at which they exit through each "outgoing" surface. Different discretization methods use different approximations and thus produce different solutions. Since the advent of large scale scientific computing, many such approximations, here called spatial discretization methods, have been formulated in order to not only allow the transport equation to be solved, but to do so in a computationally efficient manner. This level of efficiency usually requires that the cell length be greater than the mean free path of the particles traveling through it, a condition known as being "optically thick". The methods employed here go as far as to use cell lengths approaching the diffusion length for problems with sufficiently high scattering ratios, a condition known as "the diffusion limit." (The scattering ratio is the probability that a neutron-nucleus in-

teraction is a scattering event.) In the diffusion limit, transport solutions satisfy diffusion problems. Many of the methods considered in this work perform well in the diffusion limit, which means they give discrete solutions that satisfy discrete diffusion problems, even with cells that are extremely optically thick.

The methods described in this review are all deterministic, and are designed to generate accurate solutions both fro optically thin problems and for optically thick, diffusive problems. Therefore, it is important to first define the conditions constituting an optically thick, diffusive environment. Optically thick implies that the volume under study is many mean free paths thick.

The UCB methods described here are among the newer of a multitude of methods created with the intent of producing accurate transport solutions efficiently. The major methods currently in use in the main particle transport code developed and used at Texas A&M, PDT, include the following major groups: finite-element methods, long characteristics (LC) methods, linear discontinuous (LD) methods, diamond difference methods, and corner balance methods. This list represents a good sampling of the methods used across the field of particle transport, although many more exist. Each method brings with it unique advantages and disadvantages relating to the mathematics used to approximate unknown quantities needed to "close" the system and generate a solution. However, important to note about all above methods is that they perform well in one-, two-, and three-dimensional problems, and that the most reliable methods subdivide the cell and perform particle balance calculations on each subcell, thus simulating a finer mesh. Thus, any new method should have both of these qualities.

Many spatial discretization methods exist that can provide accurate solutions if the spatial mesh is sufficiently fine. However, the computational cost of transport problems, whose solutions depend on three position variables, two direction variables, one energy variable, and one time variable, would often be prohibitively large if such a fine mesh were to be employed. That is, if a spatial discretization method requires a fine mesh for a given class of problems,

the problems may not be solvable with existing computational resources. Obtaining an accurate solution on a coarser mesh is significant because it makes it possible to solve important problems of practical interest. Thus, computationally efficient spatial discretization methods should perform reasonably well with a coarse mesh discretization. This is another desirable quantity in any new method.

This paper seeks to provide a qualitative discussion of the spatial discretization methods currently in use for particle transport simulations, and subsequently to detail the genesis and development of the upstream corner balance (UCB) family of methods, a spatial discretization method believed to exhibit the above qualities.

## Summary of Currently Employed Methods

*Finite Element Methods*

Finite element methods construct a system of equations for each cell which is then solved for the flux components in the cell given the cell boundary conditions. First, a set of basis functions are selected which describe the spatial dependence of the angular and scalar fluxes when they are decomposed into a number of components. Then, a set of linearly independent weight functions are selected for each cell, and the transport equation is multiplied by each of these weight functions and integrated over the cell volume. This results in a set of linearly independent equations for each cell which is solved by the vector of flux components.

Finite element method fall in to two main subgroups, discontinuous and continuous. In continuous finite element methods, however, the weight functions used are continuous, although not necessarily nonzero, across the entire volume of interest. Continuous finite element methods are susceptible to unphysical oscillations in their solutions and are rarely used for particle transport. They will not be discussed further here.

In discontinuous finite element methods, on the other hand, the basis and weight function are assumed to be continuous within each cell, but are generally discontinuous at the cell boundary. Behaviorally speaking, discontinuous finite element methods fall into two distinct

groups: : those which fail because their leading-order solutions solve equations having little or no physical meaning, and those which are more robust, solving the correct, properly discretized diffusion equation in that region. However, these reasonably-behaving methods suffer from several defects, and are in need of significant modification to eliminate these defects. Also, it should be noted that multidimensional applications of DFEMS in the thick diffusion limit have not behaved particularly well. However, given the benefits of proper problem section and continued research into eliminating known defects, DFEMs still have great potential in particle transport simulation. [1] [2]

*Long Characteristics Methods*

Long characteristics (LC) methods are deterministic methods which operate by finding the solution along parallel lines which pass through the problem volume for each angular direction. Given the approximate spatial distribution of the total source rate density, the method solves the transport equation for the angular flux along each line. Since the LC method is a discrete ordinates method, the many directional angular fluxes are collapsed into a single cell-average scalar flux using quadrature sets, or weighted summations to approximate the integrals. [3]

Interestingly, it has been observed that in diffusive regions with optically thick spatial cells, most characteristics methods of note behave almost identically to corresponding discontinuous finite-element methods. Thus, like DFEMs, two distinct subgroups of characteristics methods exist: those which fail because their leading-order solutions solve equations having little to no physical meaning, and those which are more robust, solving the correct, properly discretized diffusion equation in that region. In addition, it was noted that characteristics methods which ignored apparently useful information on the surfaces of cells often were more robust than most commonly used characteristics methods. However, although these methods were robust, the solutions generated were often extremely inaccurate for certain cell geometries. [4]

*Linear Discontinuous Methods*

The linear discontinuous method is a commonly used spatial discretization method having its origins in finite-element methods. It combines the use of discrete ordinates to discretize the angular dependencies with an assumption of linear spatial dependence of the angular flux between boundaries. Equations for the coefficients in the linear expansion are obtained as described above, by spatially integrating the transport equation after multiplying by different weight functions.

Cell-edge fluxes generated by the LD method are very accurate on all cells except those containing boundary layers not resolved by the spatial mesh. In addition, the LD method is surprisingly accurate for interior cells not directly abutting defined boundaries. A major objection to the LD method is its large storage requirement compared to other similar methods, namely the diamond difference method. This, however, is offset by the fact that DD methods must employ a very fine mesh in unresolved boundary layers in order to achieve similar levels of accuracy to the LD method. [5]

*Diamond Difference Methods*

The diamond difference method is another commonly used spatial discretization method. It combines the use of discrete ordinates to discretize the angular dependencies with the flux spatial dependence in each cell defined by edge values and a center average value. The closure approximation is that the cell averaged flux is a simple arithmetic average of any two opposite-surface averaged fluxes.

Idiosyncrasies of the diamond difference method mainly concern prescribed boundary conditions and boundary behavior. For the diamond difference method, it is known that if prescribed incident fluxes are isotropic, then cell-edge fluxes will be accurate in optically thick, diffusive problems. However, for anisotropic incident boundary fluxes, inaccuracies in the cell-edge and cell-average fluxes can occur in the thick diffusion limit. Also noteworthy is the breakdown of DD method accuracy in cases where volume boundary layers are not

resolved by the spatial mesh. In non-diffusive problems, DD can be highly susceptible to strong unphysical oscillations. [6] [5]

*Corner Balance Methods*

Corner balance methods are the group of spatial discretization methods of primary focus to this review. Corner balance methods approximate a finer mesh than is used by further subdividing each cell into subcell volumes known as corners. Then, depending on the corner balance method employed, specific assumptions and algorithms are employed to solve for the scalar flux in each corner within a cell. These fluxes are then volume-averaged, giving the cell average scalar flux. A more in-depth discussion of corner balance methods, specifically the genesis of and ongoing research in upstream corner balance (UCB) methods, is found in the next section.

**Upstream Corner Balance Methods**

Upstream Corner Balance (UCB) methods employ a combination of exact equations (the neutron transport equation) and approximate equations to find the unknowns outgoing angular fluxes at each cell, thus "closing" the cell using only "incoming" or "upstream" values. UCB operates by further subdividing the cell into corners, defined here as the volume enclosed by a cell vertex, the adjacent cell face midpoints, and a defined cell center. An example of this cellular subdivision in two dimensions can be seen in Figure I.1.

We begin with the following steady-state transport equation:

$$\vec{\Omega} \cdot \vec{\nabla}\psi + \sigma_t(\vec{r})\psi(\vec{r},\vec{\Omega}) = \frac{1}{4\pi}\left[\sigma_t(\vec{r}) - \sigma_r(\vec{r})\right]\phi(\vec{r}) + S(\vec{r},\vec{\Omega}), \quad \vec{r} \in V \tag{I.1}$$

where $\vec{\Omega}$ is the vector representing the direction under consideration, $\psi(\vec{r},\vec{\Omega})$ is the angular flux, $\phi(\vec{r})$ is the scalar flux, $\sigma_t$ is the total interaction cross section, $\sigma_r$ is the removal cross

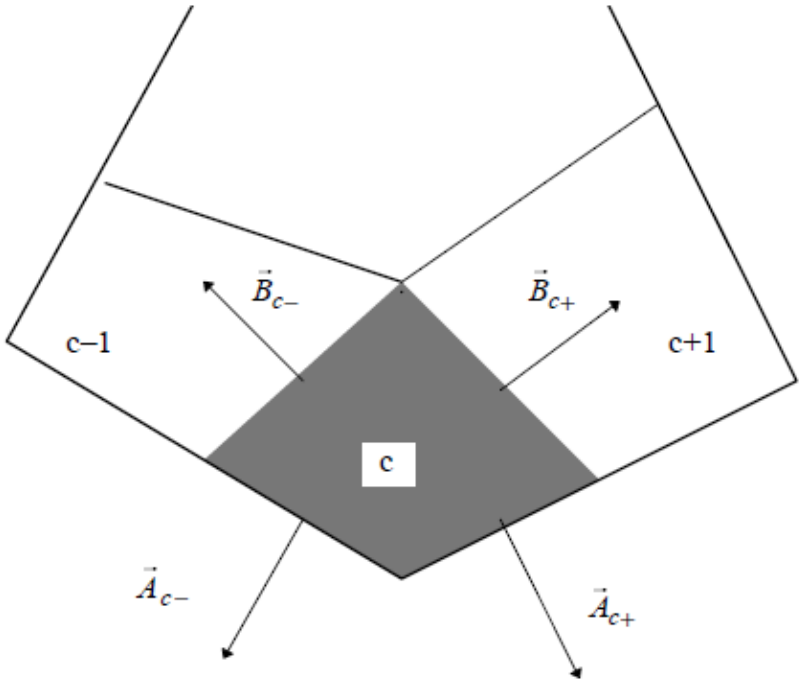



Fig. I.1. Corner c and its neighboring corners in a polygonal cell. Also shown are outward normal vectors, each with a magnitude of its respective edge length

[7]

section, and $S(\vec{r},\vec{\Omega})$ is the position and direction dependent source-rate density. Integrating it over an arbitrary corner $c$, we obtain a conservation equation for corner $c$:

$$\vec{\Omega}\cdot\left[\vec{A}_{c+}\psi_{c+}+\vec{A}_{c-}\psi_{c-}+\vec{B}_{c+}\psi_{c+1/2}+\vec{A}_{c-1/2}\psi_{c-1/2}\right]+\sigma_{t,c}V_c\psi c=V_c\left(\frac{1}{4\pi}\left[\sigma_{t,c}-\sigma_{r,c}\right]\phi_c+S_c\right) \tag{I.2}$$

where $\vec{A}_{c+}$, $\vec{A}_{c-}$, $\vec{B}_{c+1/2}$, and $\vec{B}_{c-1/2}$ are the normal vectors seen in Figure I.1, each of which has a magnitude equal to the length of its respective edge, and $\vec{\psi}_{c+}$, $\vec{\psi}_{c-}$, $\vec{\psi}_{c+1/2}$, and $\vec{\psi}_{c-1/2}$ are the average intensities over each edge. This is the equation which UCB populates and solves at each corner in the cell, sweeping across the cell in the downstream direction. This equation contains no approximations, but in order to calculate the five averaged intensities (four edge-averaged and one corner volume-averaged) using only known upstream values, we employ a number of approximations.

If the intensity on a cell-surface edge is unknown, it is approximated with the corner-averaged unknown; otherwise it is a known incident flux:

$$\psi_{m,c\pm}=\begin{cases}\psi_{m,c} & \vec{A}_{c\pm}\cdot\vec{\Omega}_m>0\\ \psi^{inc}_{m,c\pm} & \vec{A}_{c\pm}\cdot\vec{\Omega}_m<0\end{cases} \tag{I.3}$$

At this point, the UCB algorithm diverges from that of its progenitor, CB-step. In CB-step, the intensity on a cell-interior corner approximated as the averaged unknown from the corner upstream of the edge:

$$\psi_{m,c+1/2}=\begin{cases}\psi_{m,c} & \vec{B}_{c+}\cdot\vec{\Omega}_m>0\\ \psi_{m,c+1} & \vec{B}_{c+}\cdot\vec{\Omega}_m<0\end{cases} \tag{I.4}$$

However, in UCB, the intensity on a cell-interior edge contains additional terms in its approximation:

$$\psi_{m,c+1/2} = \begin{cases} \psi_{m,c} + \frac{1}{2\sigma_t}\left(Q_{c+1} - Q_c\right) + [T]_{m,c\to c+1} & \vec{B}_{c+} \cdot \vec{\Omega}_m > 0 \\ \psi_{m,c+1} + \frac{1}{2\sigma_t}\left(Q_c - Q_{c+1}\right) + [T]_{m,c+1\to c} & \vec{B}_{c+} \cdot \vec{\Omega}_m < 0 \end{cases} \tag{I.5}$$

Here $\vec{\psi}_c$ and $\vec{\psi}_{c+1}$ are corner volume-averaged angular fluxes and $Q_c$ and $Q_{c+1}$ are the total source rate densities for a corner and its neighbor. In this case, the term denoted $T$ represents the modifiable nature of UCB. By changing the form and dependencies of $T$, we are able to implement different versions of UCB. The $T$ term can have many dependencies, but in most cases employs incident flux values to establish a more realistic flux gradient across the corner, resulting in more realistic solutions. The $T$ term is conducive to current research due to the simplicity with which new UCB methods can be added once the initial framework is in place. [7]

The discussion in this chapter establishes the significance, background, and basis for this research.

# CHAPTER II

# METHODOLOGY

In order to accomplish the long-term goal of this research, namely determining methods by which particle transport phenomena may be accurately simulated in certain circumstances for which have historically been problematic, four specific aims had to be accomplished. First, we must successfully implement different UCB methods in the massively parallel transport code PDT. Second, once UCB methods are implemented in PDT, the accuracy of these methods must be evaluated by running simulations of many test problems, the solutions of which are known and verified. Third, the strengths and weaknesses of UCB methods will be analyzed and the question will be posed as to whether or not a new formulation might result in a new UCB method that has significant advantages over previous UCB methods and over competing DFEMs. Finally, at the conclusion of this research, recommendations will be made as to whether any of the tested UCB methods should be employed in production codes at national laboratories, in light of their strengths and weaknesses relative to the competing DFEMs.

## Implementation

Before any meaningful performance information can be obtained for any new algorithm, it must be implemented in an appropriate simulation engine. In this case, we employ the massively-parallel particle transport code PDT, developed over the last 16 years by a team at Texas A&M University. PDT was created as part of an effort to develop strategies to produce discrete-ordinates solutions to the transport equation in very short time frames on computational platforms relevant to the Department of Energy's Accelerated Strategic Computing Initiative (ASCI). [8] [9]

In order to implement the desired changes in PDT, the PDT source code itself had to be augmented and edited. Rather than completely implement UCB methods from nothing, the

infrastructure of a similar method already in use in PDT was edited to employ the UCB algorithm instead. The "scaffold" files selected define the two-dimensional
Corner Balance-Step method described in Chapter I.

In the course of this project, five files were created. The first newly created file, UCB_Method_2D.cc, contains the entire UCB algorithm for two-dimensional problems, as detailed above in equations I.3 and I.5.

The second new file, UCB_Method_2D.h is the basic header file for all UCB methods. In the C++ language, the practice of placing oft-used lines of code such as forward class, subroutine, variable declarations and module inclusions in a header file saves space in the source code file and saves the time of including these statements in every file employing them. In this case, UCB_Method_2D.h contains code irrelevant to the mechanics of the method that would only serve to clutter the main source code file.

The remaining three new files, UCB_Method_2D_97.h, UCB_Method_2D_RJ.h, and UCB_Method_2D_CB.h each represent a different version of the UCB algorithm. The distinction between the three methods lies in the definition of the aforementioned "T term" from equation I.5. In this case, the header file is employed to contain code unique to each method, allowing the same .cc file to be used every time, regardless of the UCB method selected. At the outset of this research, two T terms were selected for study. The first of these, UCB_Method_2D_97.h, defines T as it was defined by Adams in 1997, according to equations II.1 and II.2 below:

$$[T]_{m,c\to c+1} = \alpha_{m,c\to c+1}\frac{\vec{A}_{c-}\cdot\vec{\Omega}_m}{\sigma_{t,c}V_c}(\psi_{m,c}-\psi_{m,c-}) \tag{II.1}$$

$$\alpha_{m,c\to c+1} = \frac{3(\vec{B}_{c\pm}\cdot\vec{\Omega}_m)^2 + 4\vec{B}_{c\pm}\cdot\vec{\Omega}_m\sigma_{t,c}V_c + \alpha_0 4(\sigma_{t,c}V_c)^2}{2(\vec{B}_{c\pm}\cdot\vec{\Omega}_m)^2 + 2\vec{B}_{c\pm}\cdot\vec{\Omega}_m\sigma_{t,c}V_c + 4(\sigma_{t,c}V_c)^2} \tag{II.2}$$

where $\alpha_0 = 0.455$ and the other variable definitions are consistent with those in Chapter I. The second header file, UCB_Method_2D_RJ.h, defines a zero-value T term. While still

in the implementation process, such a definition was selected for testing in order to have a reference point for the effect of each T term on UCB method results. [7]

Based on the great similarities between the Corner Balance-Step (CB-STEP) spatial discretization method and UCB methods, it was decided to implement UCB in such a way that CB-STEP and UCB methods could employ the same .cc file. Therefore, the third header file, UCB_Method_2D_CB.h, is invoked when a CB-STEP method is selected, although as their progenitor, it is now encoded as a member of the UCB family of methods.

Finally, the central PDT program files were edited to account for the introduction of these new files to the PDT source code. Edits made to this file were done in close cooperation with the research staff at the Center for Large Scale-Scientific Simulation, and are of a technical nature not important to the methodological discussion presented here.

**Simulation**

The necessary simulations for this phase of the project were carried out on the 30 node, 248 core computational cluster operated by the Texas A&M Department of Nuclear Engineering. In order to facilitate the mass generation of data necessary for this part of the project, a perl script was constructed with the capability to run all the input files for a given discretization type in a directory. In addition, to facilitate side-by-side comparisons, if provided with other spatial discretization methods as arguments, this script, named prefontaine.pl, is able to edit and rerun each input file such that each test problem is simulated for all desired spatial discretization methods. After it completes simulating all test problems in the directory for a given discretization type, the script will optionally purge the directory of all temporary input files, leaving only the output files, which it will subsequently rewrite if run again. The final capability of prefontaine.pl is the conversion of all output files to plots of the flux distribution in the simulation volume. This is accomplished through the automated use of a utility known as pdt2silo and a program known as VisIt. Developed at Texas A&M as part of the PDT distribution, pdt2silo performs the simple but important task of converting the PDT output files to the SILO format, which VisIt, a visualization software developed at

Lawrence Livermore National Laboratory, is then able to convert to a plot in the form of an image file. The use of prefontaine.pl greatly decreases time spent producing extra input files, and thus, allows for the generation of much more test problem data than would be possible otherwise.

### Verification

Central to a well-designed and well-implemented code are two important qualities: the discretizations' proper approximation of the underlying mathematical model (the transport equation, in our case) and the code's accurate implementation of the discrete equations and their solution procedures. The verification process for a newly implemented method such as UCB can be broken down into two distinct phases: code verification and solution verification. Code verification consists of developing tests problems which expose differences between the code's output and the discrete solution that it should produce if it were bug-free. Thus, the central aim of this step is locating and repairing errors in the source code. The main goal of solution verification is the quantification of the numerical error arising from the discretization and the (iterative) solution processes, for some quantities of interest in some problems of interest. This numerical error is a consequence of the approximations made in the closure equations, in the case of UCB. By quantifying the numerical error in an algorithm, it can be considered when the algorithm's results are examined or utilized elsewhere.

To validate a model, the results from a code that employs the model are compared with experimental or known data in order to quantify the degree to which the model is representative of the real world process being modeled. Validation of the linear Boltzmann equation for particle transport was outside the scope of this work. That is, the present work addresses code and solution verification, but not model validation.

At present, verification efforts have focused upon the UCB˙97 method, as it represents the best chance of improvement over current methods.

*Code Verification*

To verify that this implentation of UCB was in accordance with the original UCB algorithm, simple test problems of known solutions were simulated using UCB. These problems were simulated for the same simulation volume divided into either a single cell or sixteen (4x4) cells. Simulations were performed both with and without scattering. A graphic representation of these problems and their intended solutions is seen in Figure II.1.

The first problem shown in Figure II.1 is a zero-value problem. If the UCB method has been implemented according to the original algorithm, a simulation volume containing no distributed source and surrounded by vacuum boundary conditions will have a scalar flux of zero at all points in the volume.

The second problem showin in figure II.1 , is a constant-solution problem. If the UCB method has been implemented according to the original algorithm, a simulation volume containing a distributed source and surrounded by incident scalar fluxes of magnitude $\phi = Q/\Sigma_a$ will have a scalar flux of the same magnitude as the boundary fluxes at all points in the volume. This follows from the discrete-ordinates equations employed by UCB:

$$\mu_n \frac{d\psi_n(x)}{dx} + \Sigma_t \psi_n(x) = \frac{\Sigma_s + \nu\Sigma_f}{2} \sum_{n'=i}^{N} w_{n'} \psi_{n'}(x) + \frac{Q}{2} \tag{II.3}$$

$$\phi(x) \approx \sum_{n'=i}^{N} w_{n'} \psi_{n'}(x) \tag{II.4}$$

When the solution is assumed constant in space and no fission is assumed, then the equations can be simplified and rearranged to obtain the constant flux solution (which also must be imposed as an incident scalar flux at all boundaries):

$$\Sigma_a \phi(x) = Q \;\Rightarrow\; \phi(x) = \frac{Q}{\Sigma_a} \tag{II.5}$$

Constant-value problems like this are important, as they allow the underlying mathematics of UCB to be partially verified with a single value.

Subsequently, another set of simple problems were simulated in order to better verify this implementation of UCB against the underlying mathematics of the method. These problems, which approximate one-dimensional slab problems, were set up according to Fig. II.3. One direction is rendered effectively infinite by making it extremely large and divided into a 3 cells. Data is only collected from the center cell on each boundary in order to remove the influence of the edges of the "infinite" direction on the scalar flux estimate each boundary. The direction of interest is given a much smaller, finite value (in this case 10 cm), and divided into N cells. Data is recorded for the estimates of the scalar flux on each edge, as given by the scalar flux calculated at the center cell on each boundary.

By simulating this problem for various N values and comparing the results against a known solution, the UCB solution's convergence upon the analytical solution can be observed and charted. For the purposes of code verification, verifying convergence is important. The analytical solution follows from the $S_2$ discrete ordinates equations (Eqs. II.3-4) when the right boundary of the slab is taken to be $\psi(10) = 0$. With no scattering, all neutrons in the material will move left to right, so $\psi_2$, the left-bound angular flux, will always be zero, and the left boundary condition becomes:

$$\phi(0) \approx \sum_{n'=i}^{N} w_{n'} \psi_{n'}(0) \;\Rightarrow\; \psi_1(0) = \frac{\phi(0)}{2} \tag{II.6}$$

When applied to the simplified $S_2$ equations:

$$\frac{-1}{\sqrt{3}} \frac{d\psi_1(x)}{dx} + \Sigma_a \psi_1(x) = 0 \tag{II.7}$$

the analytical solution obtained is:

$$\psi_1 = \psi_1(0)e^{-\sqrt{3}\Sigma_a x} = \frac{\phi(0)}{2}e^{-\sqrt{3}\Sigma_a x} \tag{II.8}$$

*Solution Verification*

The one-dimensional slab problems simulated as part of the code verification process can also be used to determine the numerical error associated with UCB. By recording the results for various N values, the relationship between mesh fineness and numerical error can be quantified. The problem detailed in Figure II.3 was simulated for various N-values, and the results were compared with the analytical solution. The relationsip between mesh fineness (as given by N) and numerical error was then plotted in Microsoft Excel. By fitting a trendline to the data, the relationship between mesh fineness and numerical error can be quantified. By determining the numerical error associated with UCB for a given mesh fineness, the accuracy of UCB for similar problems can be accurately determined.

*Other Verification*

Finally, as an additional verification of this implementation of the UCB algorithm, simple problems were simulated for multi-cell volumes in order to ensure that the method gives symmetrical solutions for test problems with symmetrical input conditions. These problems were chosen because, given symmetrical input, they produce consistent symmetrical output if a given spatial discretization method is functioning properly. Each problem was simulated for the same simulation volume divided into sixteen (4x4) cells, with and without distributed sources, with and without scattering. These problems are shown in Figures II.2. Note that expected lines of symmetry are denoted on each problem diagram.

**Comparison**

An effort was made to compare the performance of UCB with that of other methods. The group of simple problems shown in Fig. II.2. was simulated for other spatial discretization methods as well as PDT, in particular the CB-STEP and PWLD methods. A visual

inspection of the plots generated verified that all three methods preserved symmetry. The similarities observed between the output from these different methods provided an added validation for solutions generated with UCB methods because the other, older methods had also been verified for physicality.

The central part of the comparison between UCB and the other two methods was the repetition of the solution verification process for the other methods. This comparison was done for both scattering and non-scattering one-dimensional slab problems. However, in order to obtain a numerical error estimate for the scattering case, the analytical solution was approximated. Noting that all three methods were apparently converging on very close values, the values generated by each for the finest mesh tested were averaged and the result was used as a pseudo-analytical solution. The convergences of all methods to the analytical solution with increasing mesh fineness were then plotted for the all cases to provide a visual comparison of the methods. Thus, the convergence orders of all three methods were compared for the scattering and non-scattering cases, allowing conclusions to be made regarding the performance of each method under a variety of conditions.

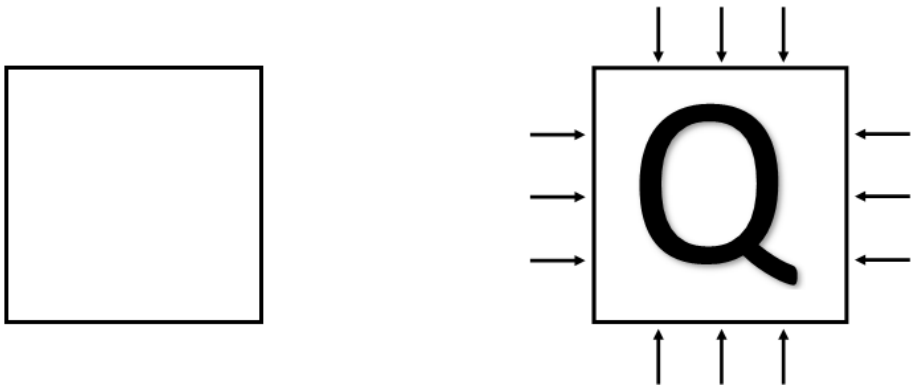



Fig. II.1. Test problems of known solutions used to verify implementation of UCB method: (a) Zero Value Problem: Vacuum boundary conditions on all sides of the simulation volume with no distributed source. $\phi = 0$ everywhere in the volume. (b) Constant Value Problem: Incident scalar fluxes on all sides of the simulation volume with a distributed source. $\phi = Q/\Sigma_a$ everywhere in the volume where Q is the source rate density and $\Sigma$ is the macroscopic absorption cross section. Note: For non-scattering problems, $\Sigma_t = 0.3$ and $\Sigma_s = 0$ and for scattering problems, $\Sigma_t = 0.6$ and $\Sigma_s = 0.3$.

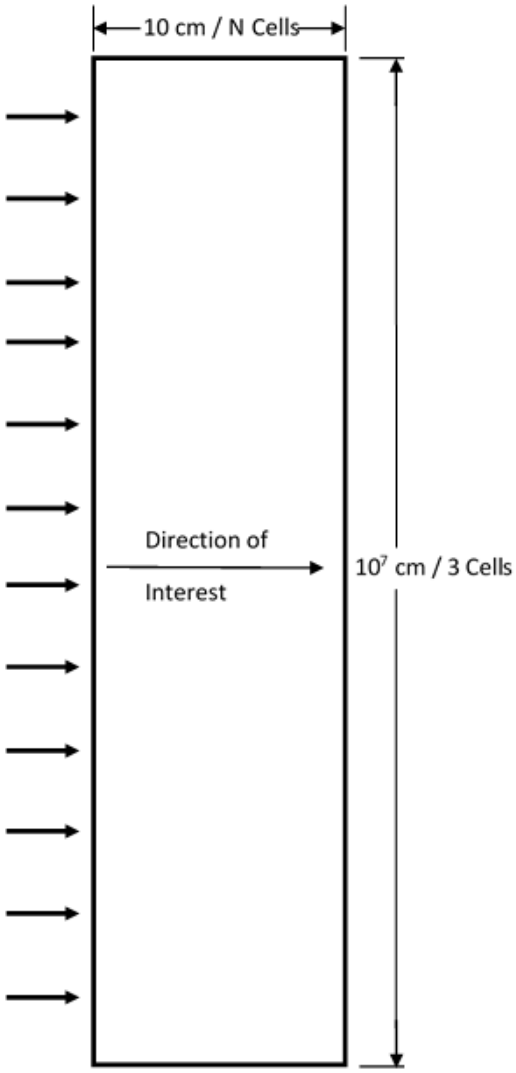



Fig. II.2. Slab problem simulated to verify and observe convergence of UCB results to analytical results. All vacuum boundary conditions except left boundary. No distributed source. Note: For non-scattering problems, $\Sigma_t = 1.0$ and $\Sigma_s = 0$ and for scattering problems, $\Sigma_t = 2.0$ and $\Sigma_s = 1.0$.

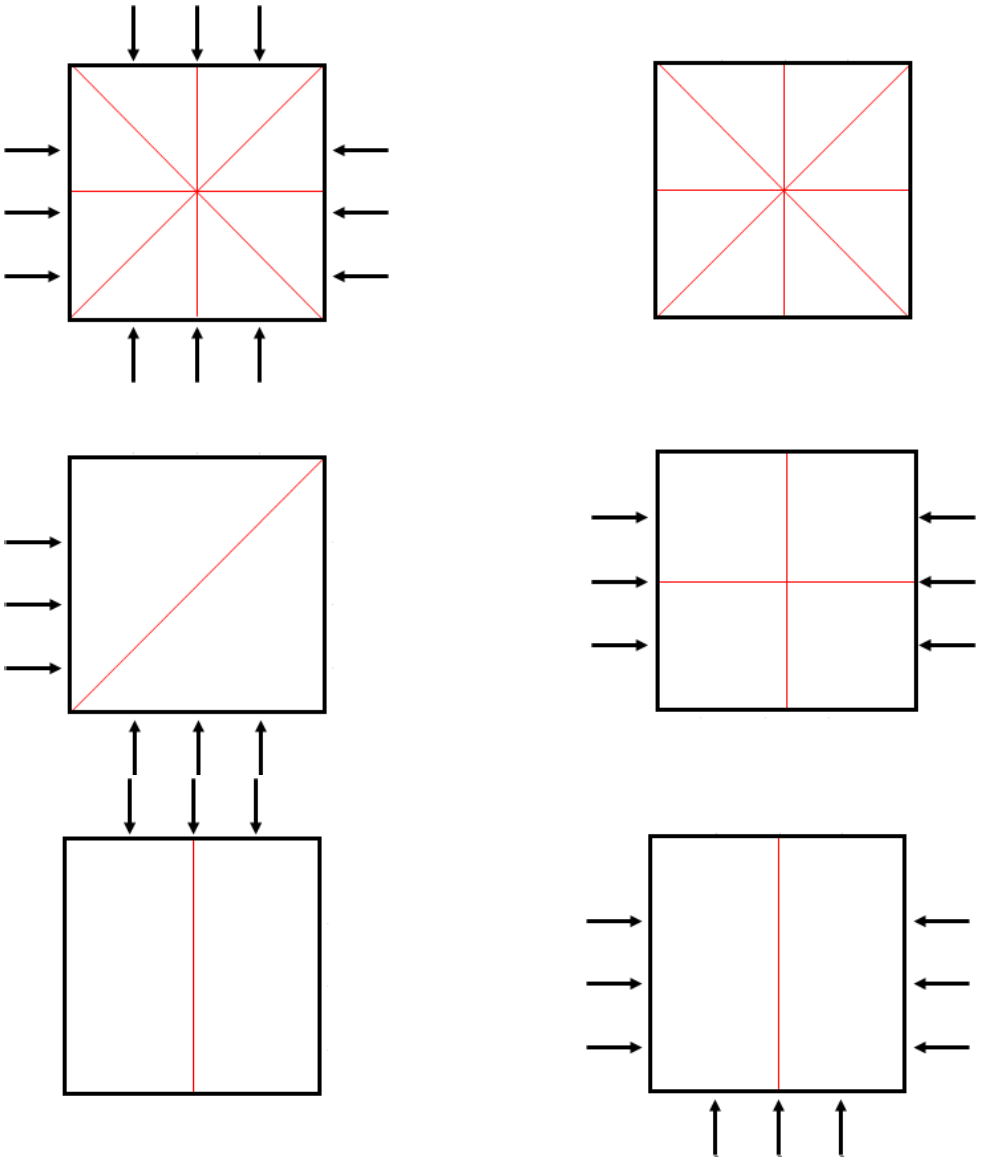


Fig. II.3. Test problems known to give symmetrical solutions used to validate implementation of UCB methods. Incident fluxes are represented by arrows, and vacuum boundaries are represented as blank. Each problem was simulated for an identical simulation volume divided into sixteen (4x4) cells. Finally, each problem was simulated with and without distributed sources. Expected lines of symmetry are noted on each diagram. Note: For non-scattering problems, $\Sigma_t = 0.3$ and $\Sigma_s = 0$ and for scattering problems, $\Sigma_t = 0.6$ and $\Sigma_s = 0.3$.

# CHAPTER III

# RESULTS

## Implementation Results

The implementation phase of this project proved to be the longest in duration and most technically challenging. Throughout this project, challenges pertaining to parallel computing and computer program architecture were encountered. In order to resolve these issues, technical advising and assistance were required from numerous individuals within the nuclear engineering department better acquainted with the parallel computing algorithms used in PDT. In addition, time constraints imposed on the project resulted in a change in project goals. Originally, the development of new spatial discretization methods was envisioned as a stretch goal, but due to the unexpected length of the implementation process, such aspirations were curtailed. In addition, also for time reasons, the decision was made to focus on evaluating the utility of the UCB_97 method in particular, although it was one of three UCB methods implemented. UCB_97 was selected because it was the original project focus and because it represented the greatest departure from current methods of the three UCB methods implemented, and thus held the most potential for improvements over previous methods. At the current time, UCB methods have been successfully implemented in PDT.

## Verification Results

*Code Verification Results*

The first problem simulated in the code verification process was the zero-value problem detailed in Figure II.1. In all cases, UCB_97 calculated a scalar flux of zero magnitude at all points in the simulation volume. Only the results for the sixteen cell scattering case are shown below, as identical results were observed for the single-cell case and scattering and non-scattering cases. A properly implemented method will demonstrate this universal agreement, regardless of simulation volume size.

The next problem simulated in the code verification process was the constant-value problem also detailed in Figure II.1. In all cases, constant values were observed across the volumes tested. The scalar flux value observed in each problem is given in Table I below. In each case, the scalar flux was constant throughout the volume, and in agreement with the analytical solution to the highest precision possible in PDT.

The final code verification problem simulated was the one-dimensional slab problem without scattering. The approach of UCB results to the analytical values is detailed in Table II.

As shown in Figures III.2 and III.3, UCB is clearly approaching the analytical values on both sides of the slab.

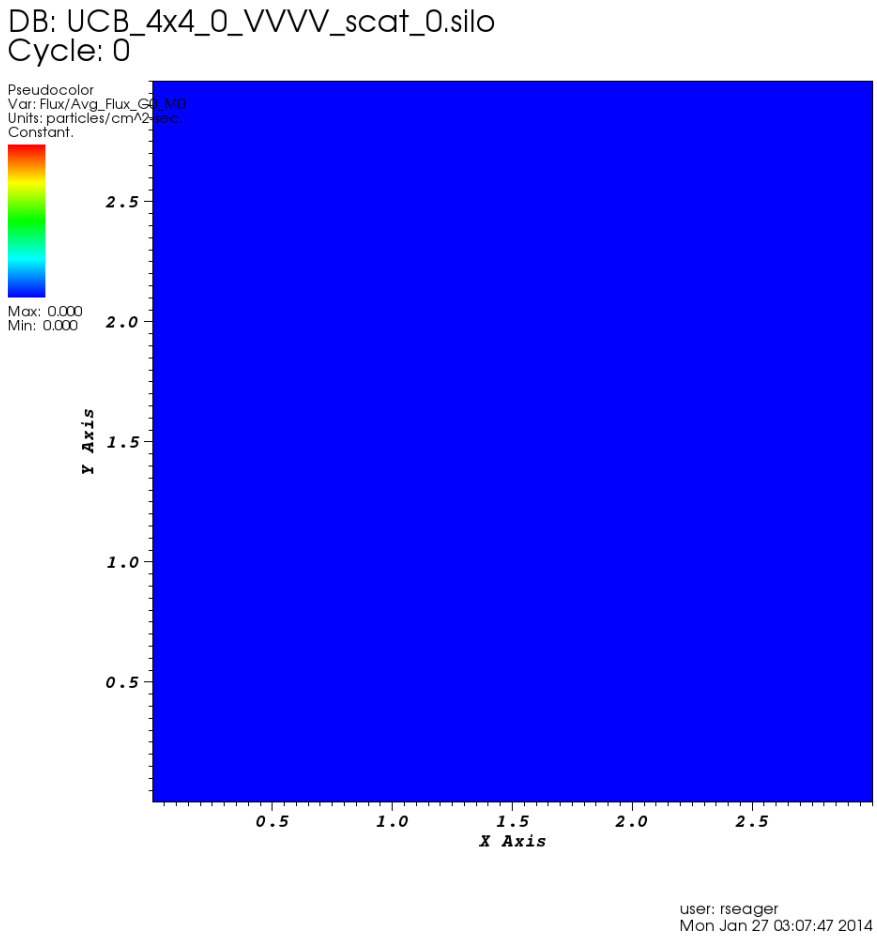



Fig. III.1. Results of zero value problem for UCB_97. Vacuum boundary conditions on all sides of the simulation volume with no distributed source. Constant value of zero throughout volume. Identical results observed in single-cell and multi-cell cases with and without scattering.

Table I. Constant Value Problem Results (Analytical Solution $\phi = 3.\overline{333}$)

| Dimension | Scattering? | Average |
|---|---|---|
| 1x1 | NO | 3.33333333333333 |
| 1x1 | YES | 3.33333333333333 |
| 4x4 | NO | 3.33333333333333 |
| 4x4 | YES | 3.33333333333333 |

Table II. UCB Approach to Analytical Values

| Cells_x | SDM/Dimens. | incident flux (n/cm^2*s) | exiting flux (n/cm^2*s) | Incident Flux Error | Exiting Flux Error |
|---|---|---|---|---|---|
| 1 | UCB_1-3 | 5.4661462045E+10 | 2.7553988235E+09 | 8.9067707591E-01 | 1.8340584720E+05 |
| 2 | UCB_2-3 | 1.0469267158E+11 | 1.5331432121E+08 | 7.9061465684E-01 | 1.0204018615E+04 |
| 4 | UCB_4-3 | 1.9393325068E+11 | 3.2852743322E+06 | 6.1213349865E-01 | 2.1767680364E+02 |
| 8 | UCB_8-3 | 3.2369101260E+11 | 9.0739202733E+04 | 3.5261797480E-01 | 5.0398483694E+00 |
| 16 | UCB_16-3 | 4.3260793151E+11 | 2.0111341270E+04 | 1.3478413699E-01 | 3.3866562759E-01 |
| 32 | UCB_32-3 | 4.8000314703E+11 | 1.5464006279E+04 | 3.9993705938E-02 | 2.9326358325E-02 |
| 64 | UCB_64-3 | 4.9454622676E+11 | 1.5057396802E+04 | 1.0907546488E-02 | 2.2613245434E-03 |
| 128 | UCB_128-3 | 4.9856479453E+11 | 1.5025795120E+04 | 2.8704109321E-03 | 1.5783056439E-04 |
| 256 | UCB_256-3 | 4.9963059504E+11 | 1.5023580700E+04 | 7.3880991010E-04 | 1.0432726036E-05 |
| 512 | UCB_512-3 | 4.9990619135E+11 | 1.5023434040E+04 | 1.8761729987E-04 | 6.7069109281E-07 |
| 1024 | UCB_1024-3 | 4.9997635633E+11 | 1.5023424603E+04 | 4.7287347810E-05 | 4.2518379177E-08 |
| 2048 | UCB_2048-3 | 4.9999406452E+11 | 1.5023424005E+04 | 1.1870955008E-05 | 2.6973701366E-09 |
| 4096 | UCB_4096-3 | 4.9999851302E+11 | 1.5023423966E+04 | 2.9739584020E-06 | 1.1817782188E-10 |
| 8192 | UCB_8192-3 | 4.9999962786E+11 | 1.5023423949E+04 | 7.4427212598E-07 | 1.0433550096E-09 |
| 16384 | UCB_16384-3 | 4.9999990692E+11 | 1.5023424032E+04 | 1.8616616602E-07 | 4.5278250414E-09 |
| analytical | | 5.0000000000E+11 | 1.5023423964E+04 | | |

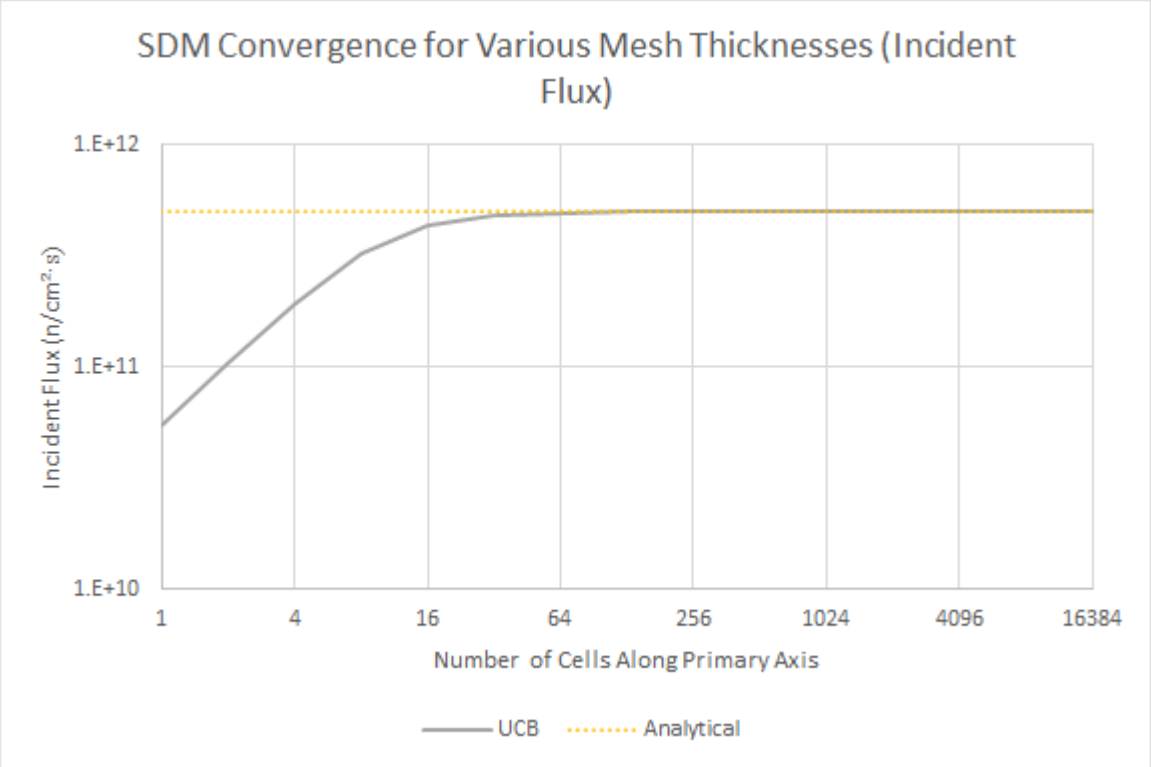



Fig. III.2. $\log_2$-$\log_{10}$ plot of the convergence of UCB to analytical value for incident flux on 1D slab with increasing numbers of cells in direction of interest.

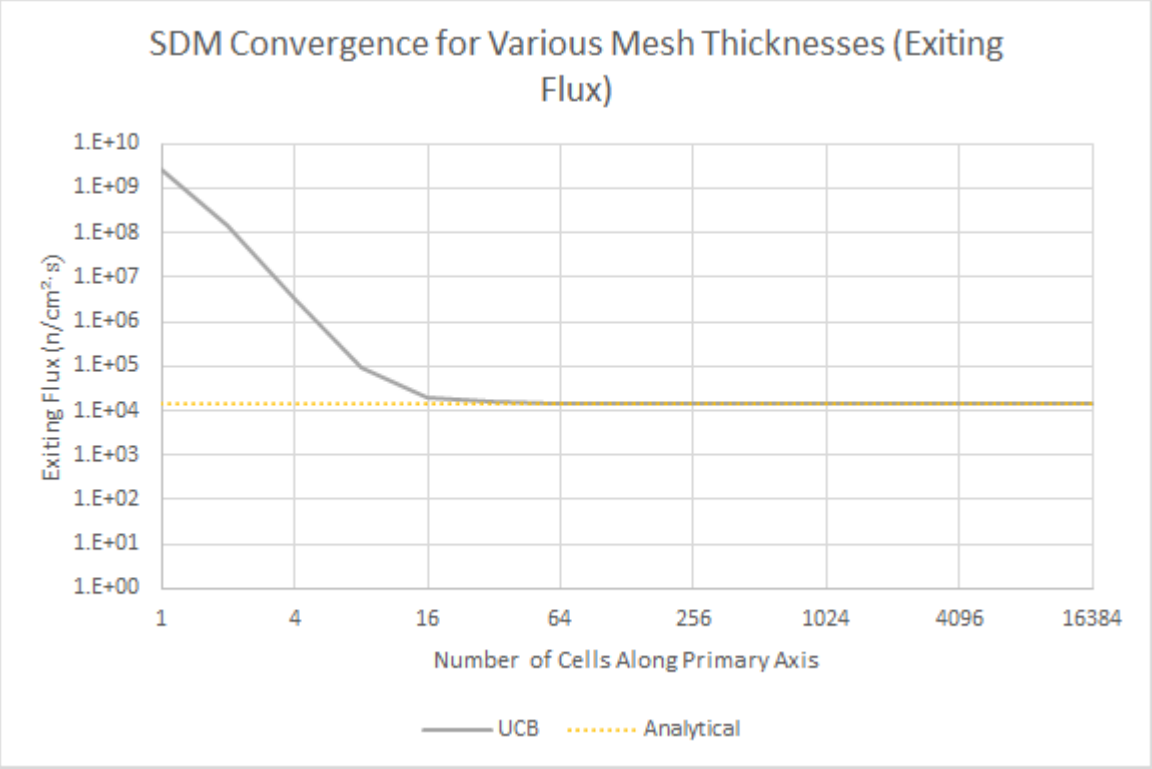



Fig. III.3. $\log_2$-$\log_{10}$ plot of the convergence of UCB to analytical value for exiting flux from 1D slab with increasing numbers of cells in direction of interest.

*Solution Verification Results*

Solution verification for UCB was accomplished by plotting the relative error data from the one-dimensional slab problem without scattering. Figure III.4 is a plot of this relationship for the incident flux, and Figure III.5 is a plot of this relationship for the flux exiting the slab. Fitting trendlines to the curves, it is observed that UCB methods have a 1.989-order convergence upon the analytical incident flux and a 4.042-order convergence upon the exiting flux. Because the incident flux is predefined and thus known with no error, the more important figure of these two is the convergence order for the exiting flux–the solution to this test problem. Therefore, UCB is best characterized as having 4.042-order convergence for these one-dimensional slab problems without scattering.

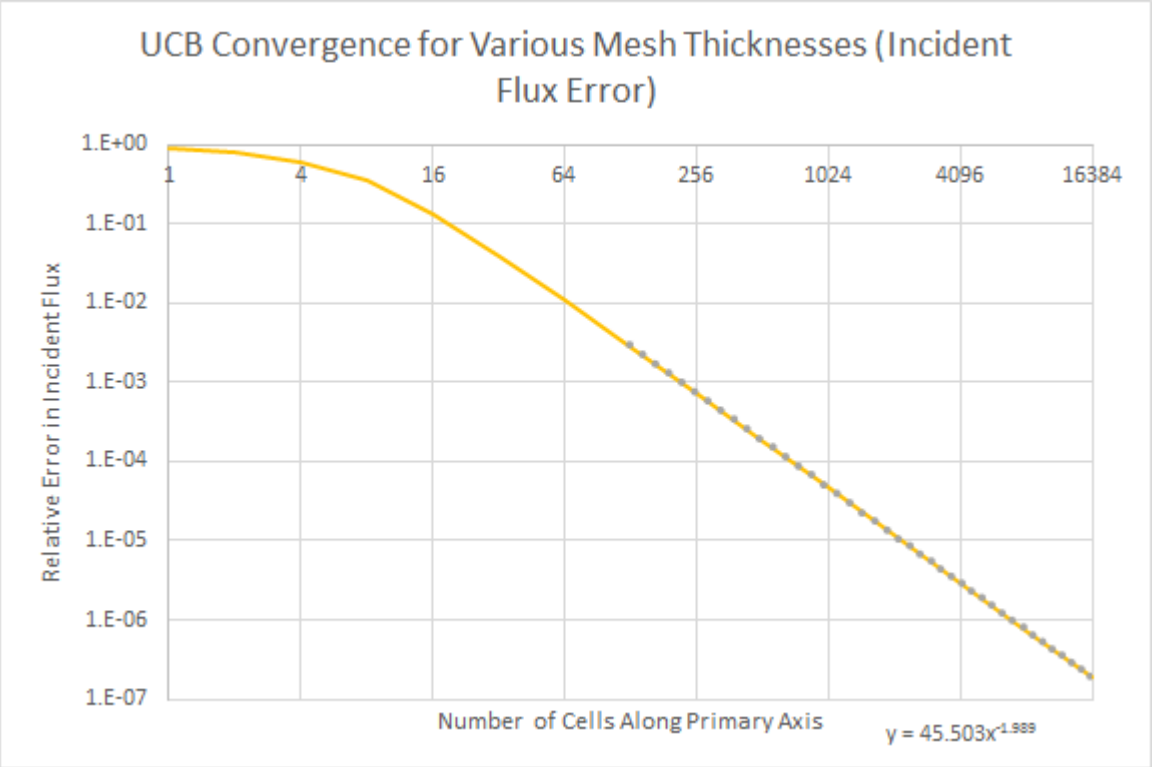



Fig. III.4. $\log_2$-$\log_{10}$ plot of the convergence of the incident flux calculated by UCB to the analytical value. Order of convergence is 1.989.

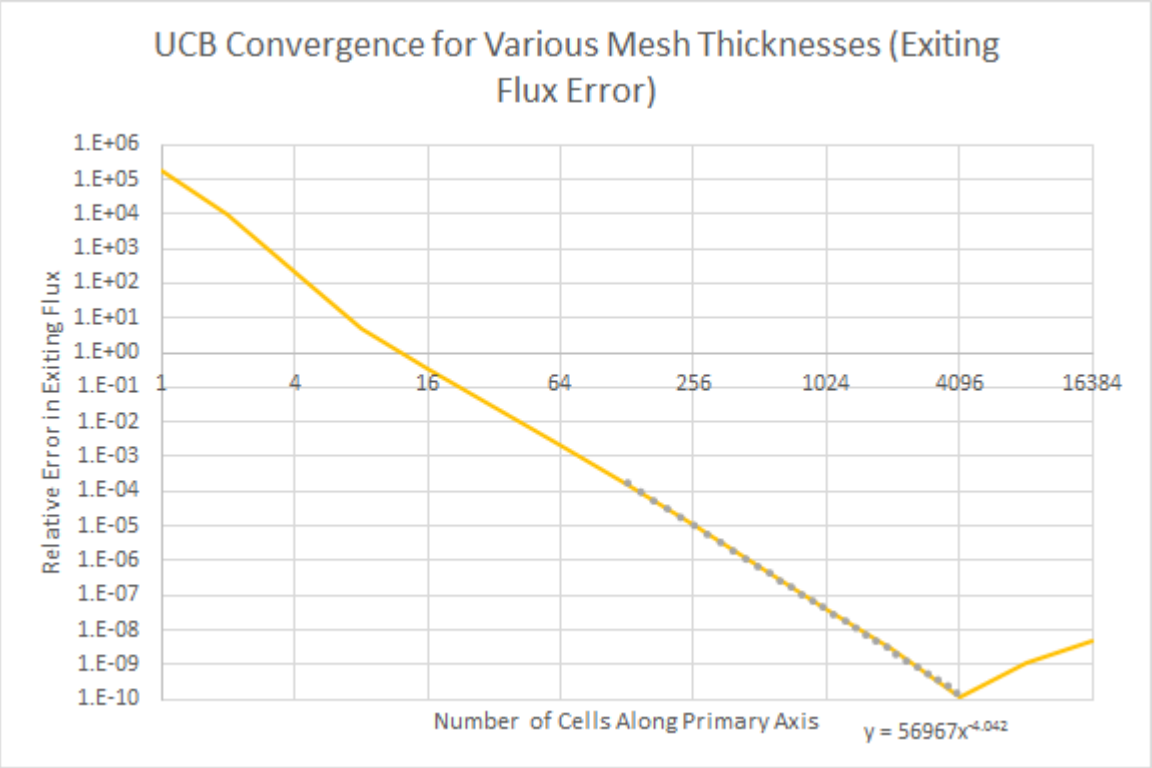



Fig. III.5. $\log_2$-$\log_{10}$ plot of the convergence of the exiting flux calculated by UCB to the analytical value. Order of convergence is 4.042.

*Other Results*

Further verification for UCB was accomplished by ensuring that UCB output for a symmetrically defined problem was indeed symmetric. This experimentally observed condition of real world situations must be present in any physically validated simulation tool. Using UCB to simulate the symmetry test problems defined in Chapter II (Fig. II.4), the following results were generated in the form of cell-averaged scalar flux plots.

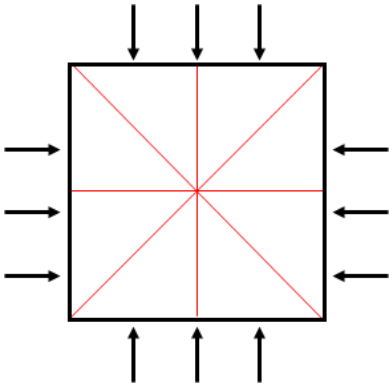
First, the symmetry problems were simulated without scattering and with no distributed source. The results are shown alongside the original problem definitions in Figure III.6.

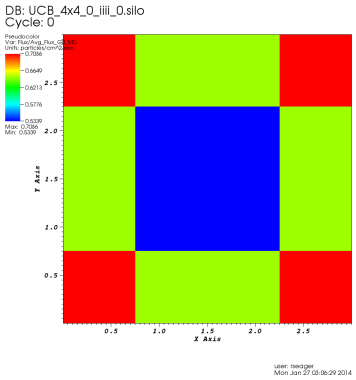
Second, the symmetry problems were simulated with scattering and still without a distributed source. The results are shown alongside the original problem definitions in Figure III.7.

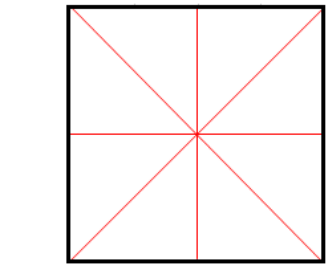
Third, the symmetry problems were simulated without scattering and with a distributed source. The results are shown alongside the original problem definitions in Figure III.8.

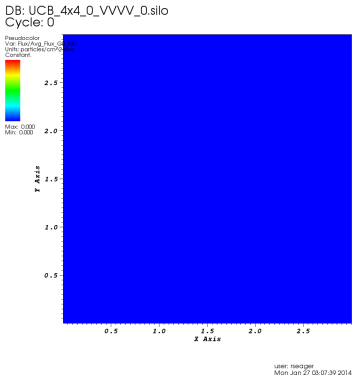
Finally, the symmetry problems were simulated with both scattering and a distributed source. These results are shown alongside the original problem definitions in Figure III.9.

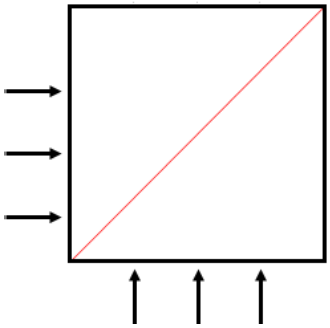
These results all very clearly show that UCB preserves symmetry for symmetrical inputs.

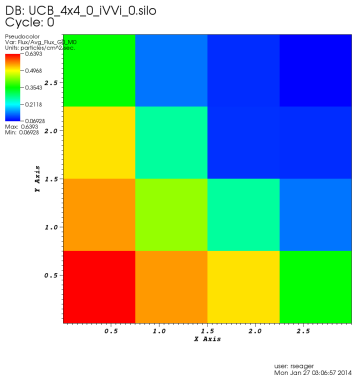



DB: UCB_4x4_0_iiii_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_0_VVVV_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_0_iVVi_0.silo
Cycle: 0
Y Axis
X Axis

(Fig. III.6.)

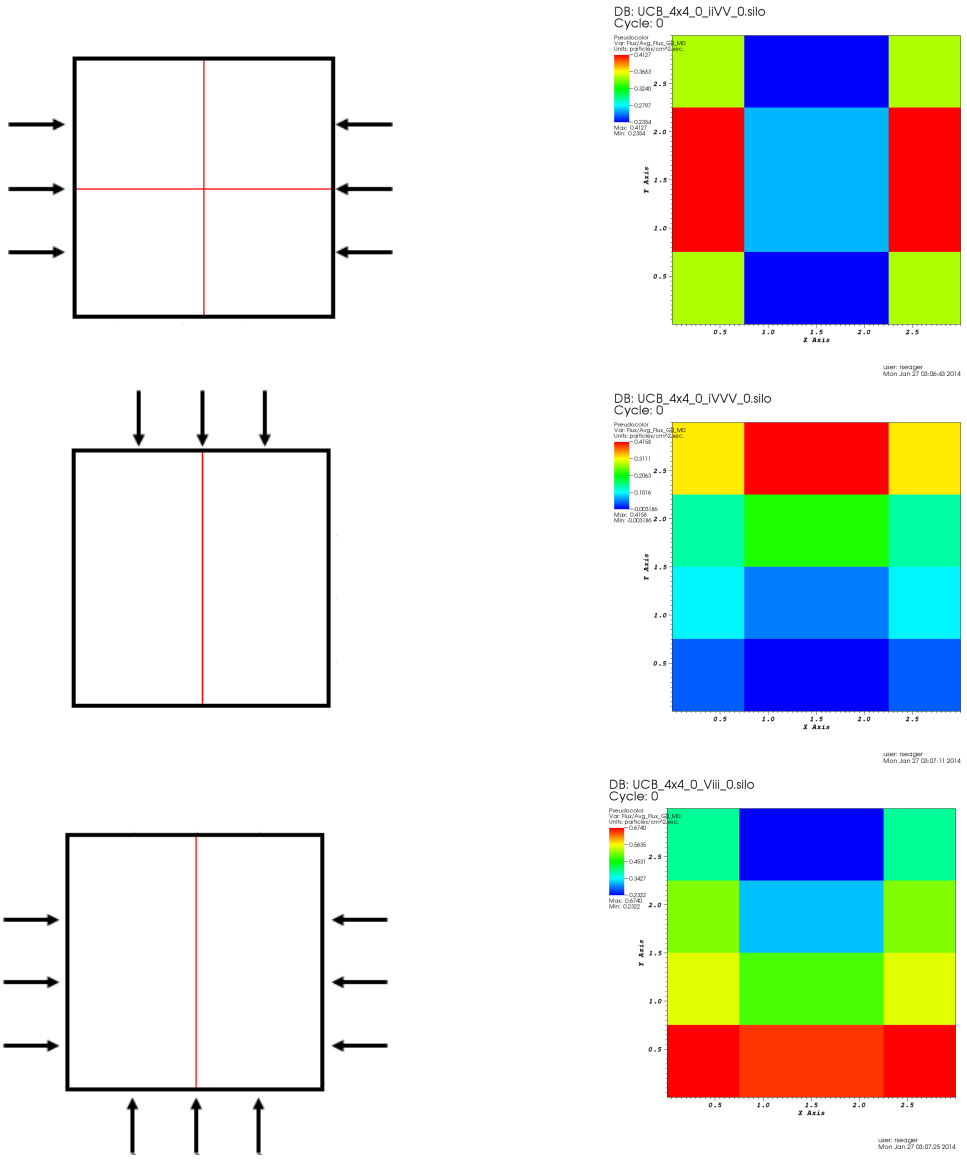
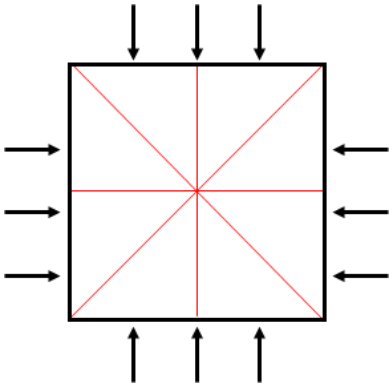
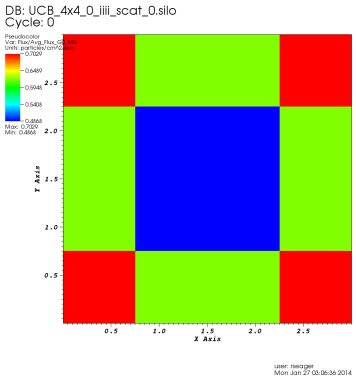
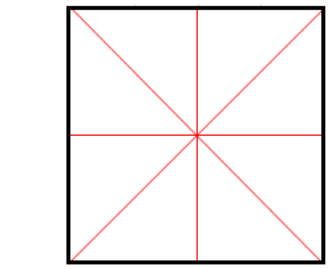
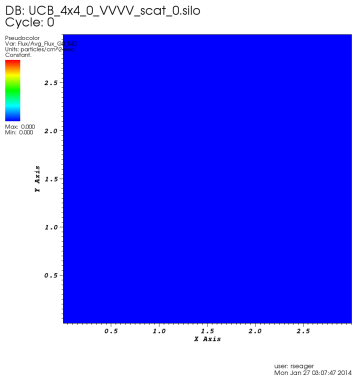


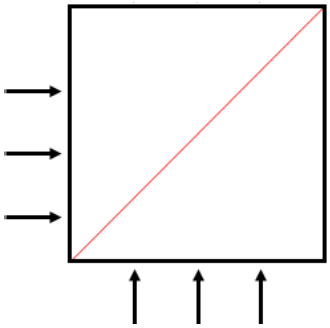

Fig. III.6. Results of test problems known to give symmetrical solutions simulated without scattering and without a distributed source. Results are adjacent to the original problem definitions. Incident fluxes are represented by arrows, and vacuum boundaries are represented as blank. Each problem was simulated for an identical simulation volume divided into sixteen (4x4) cells. Expected lines of symmetry are noted on each diagram. For these problems, $\Sigma_t = 0.3$ and $\Sigma_s = 0$.

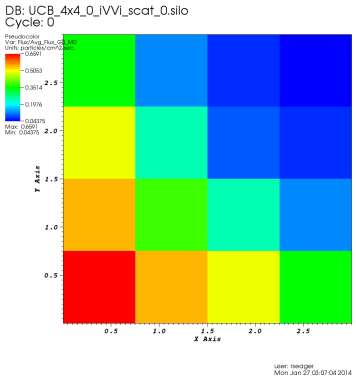



DB: UCB_4x4_0_iiii_scat_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_0_VVVV_scat_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_0_iVVi_scat_0.silo
Cycle: 0
Y Axis
X Axis

(Fig. III.7.)

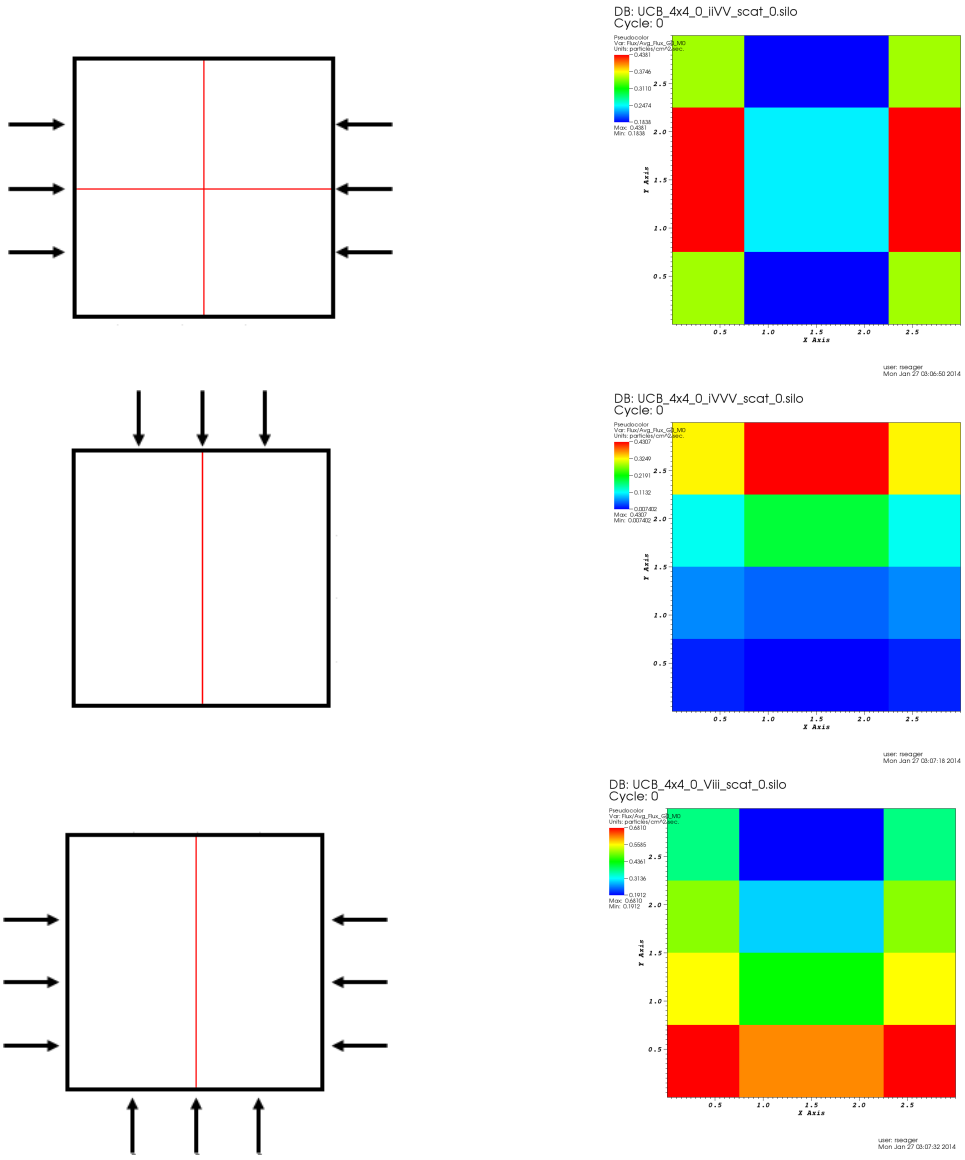
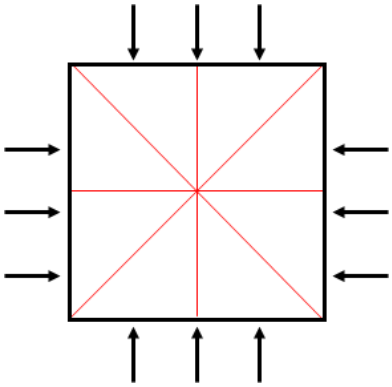
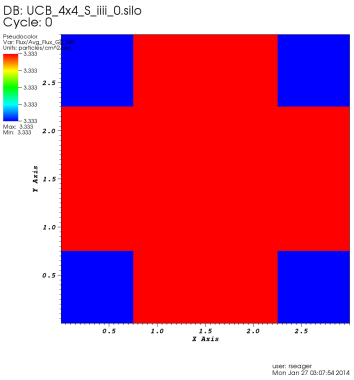
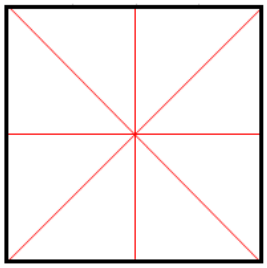
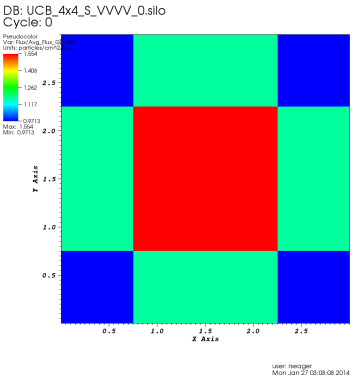


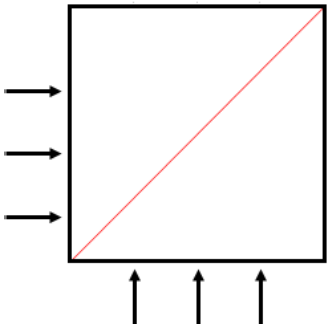

Fig. III.7. Results of test problems known to give symmetrical solutions simulated with scattering and without a distributed source. Results are adjacent to the original problem definitions. Incident fluxes are represented by arrows, and vacuum boundaries are represented as blank. Each problem was simulated for an identical simulation volume divided into sixteen (4x4) cells. Expected lines of symmetry are noted on each diagram. For these problems, $\Sigma_t = 0.6$ and $\Sigma_s = 0.3$.

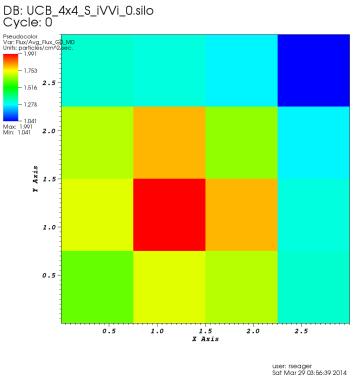



DB: UCB_4x4_S_iiii_0.silo
Cycle: 0
X Axis
Y Axis




DB: UCB_4x4_S_VVVV_0.silo
Cycle: 0
X Axis
Y Axis




DB: UCB_4x4_S_IVVI_0.silo
Cycle: 0
X Axis
Y Axis

(Fig. III.8.)

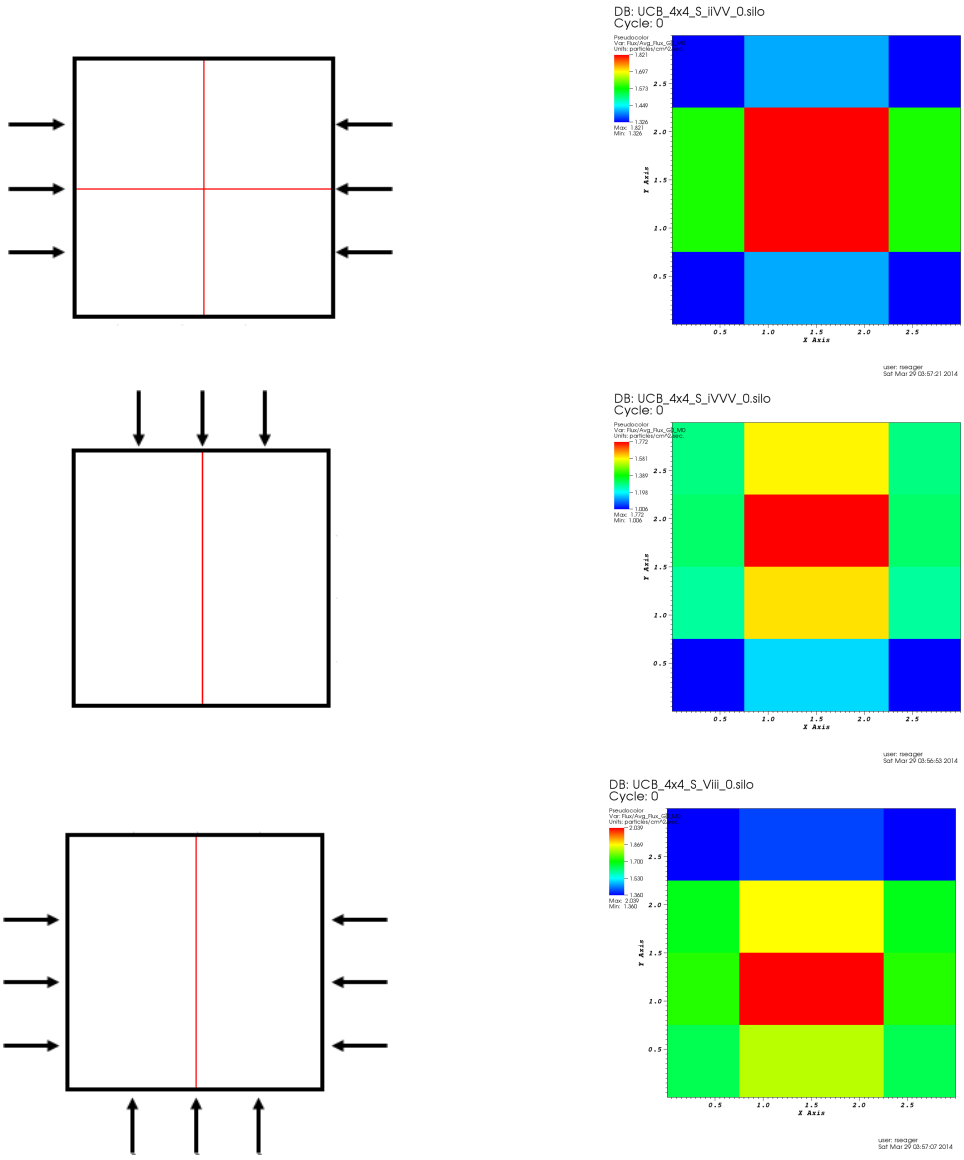
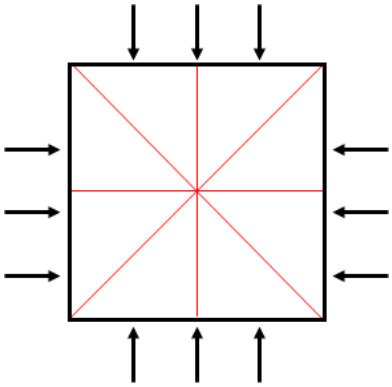
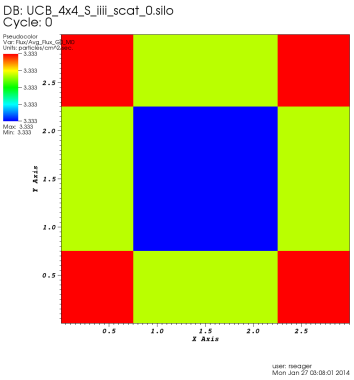
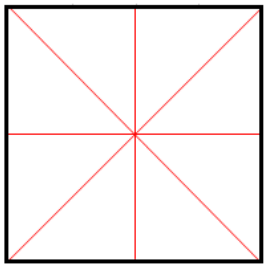
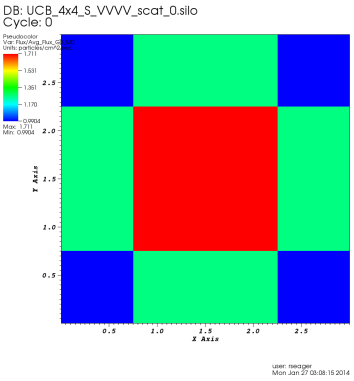


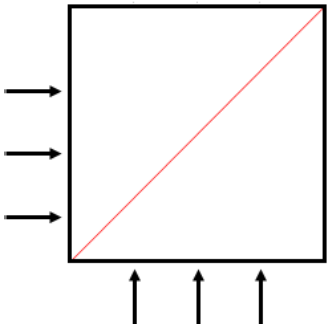

Fig. III.8. Results of test problems known to give symmetrical solutions simulated without scattering and with a distributed source. Results are adjacent to the original problem definitions. Incident fluxes are represented by arrows, and vacuum boundaries are represented as blank. Each problem was simulated for an identical simulation volume divided into sixteen (4x4) cells. Expected lines of symmetry are noted on each diagram. For these problems, $\Sigma_t = 0.3$ and $\Sigma_s = 0$.

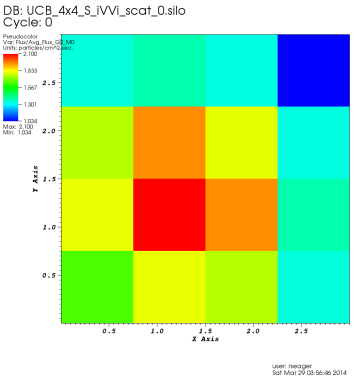



DB: UCB_4x4_S_iiii_scat_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_S_VVVV_scat_0.silo
Cycle: 0
Y Axis
X Axis




DB: UCB_4x4_S_IVVi_scat_0.silo
Cycle: 0
Y Axis
X Axis

(Fig. III.9)

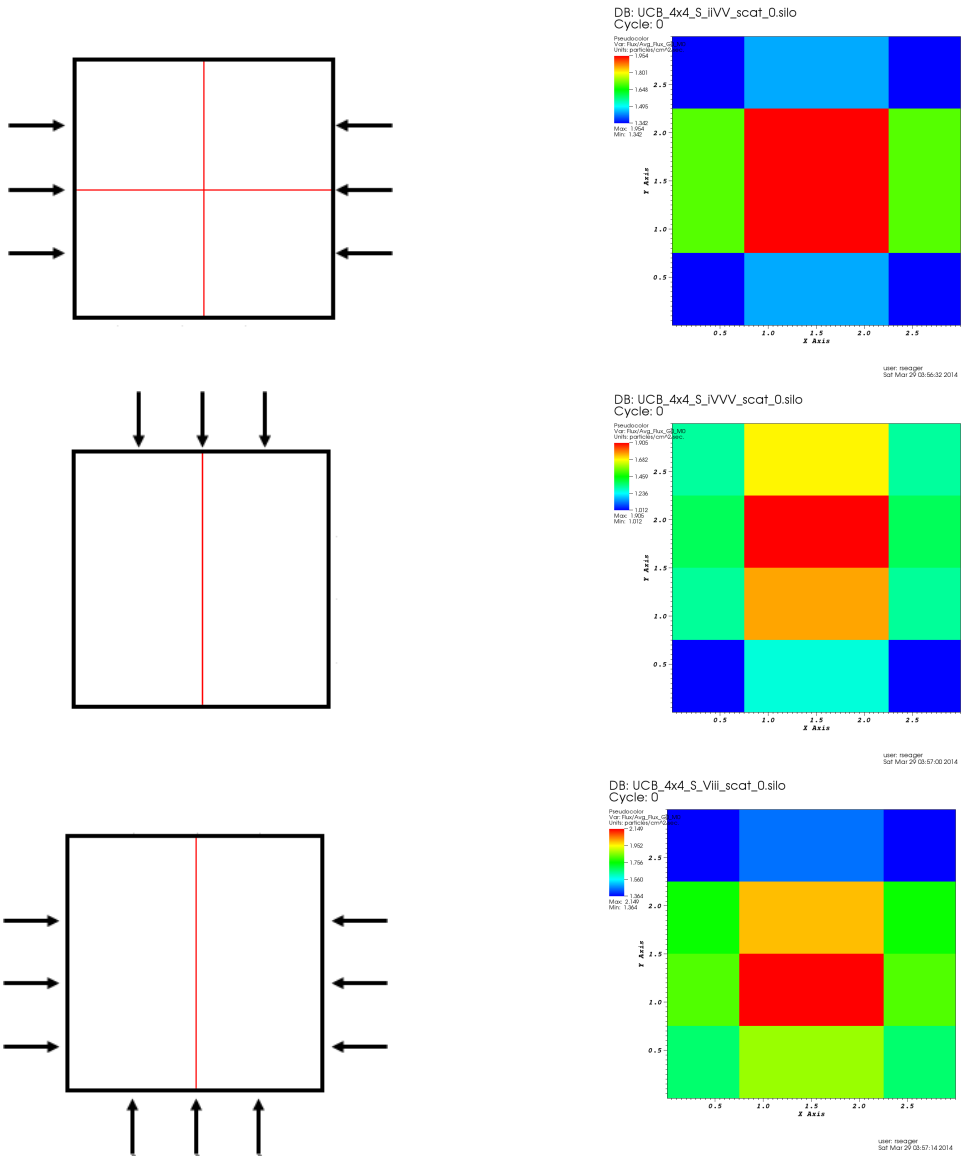



Fig. III.9. Results of test problems known to give symmetrical solutions simulated with scattering and with a distributed source. Results are adjacent to the orignal problem definitions. Incident fluxes are represented by arrows, and vacuum boundaries are represented as blank. Each problem was simulated for an identical simulation volume divided into sixteen (4x4) cells. Expected lines of symmetry are noted on each diagram. For these problems, $\Sigma_t = 0.6$ and $\Sigma_s = 0.3$.

**Comparison Results**

To compare UCB with other spatial discretization methods, the slab problem detailed in Figure II.2 was again employed. Two other spatial discretization methods, CB-STEP and PWLD, were selected for comparison. The convergence of the results of each method to analytical (for non-scattering problems) or pseudo-analytical (for scattering problems) solutions was quantified by tabulating the relationship between numerical error and the fineness of mesh as quantified by the number of cells in the direction of interest. When trendlines were plotted for each data set, data points for very coarse and very fine meshes were omitted in order to remove error and accurately determine the numerical error associated with each method.

Beginning with the non-scattering problem, Table III below shows the outputs of each of the three methods and the numerical error observed for each.

Figure III.10 shows the numerical error associated with each method's values for the incident flux as a function of the number of cells in the direction of interest.

Figure III.11 shows the trendlines derived from the data shown in Figure III.8 and their equations, which give the order of convergence for each method.

Figure III.12 shows the numerical error associated with each method's values for the exiting flux as a function of the number of cells in the direction of interest.

Figure III.13 shows the trendlines derived from the data shown in Figure III.10 and their equations, which give the order of convergence for each method.

When trendlines are calculated for this data, the orders of convergence for each method can be determined. Table IV lists the convergence orders of the incident and exiting fluxes for each method. When calculating the incident flux, UCB and PWLD share approximately 2nd-order convergence, although UCB converges slightly faster. CB-STEP demonstrates single-order convergence. However, important to note again is the fact that the incident flux is user-defined and therefore already known with no error, so the convergence order of the

exiting flux is the more important figure for characterizing each method. From this data, it is clear that when calculating the exiting flux for 1D slab problems without scattering, UCB outperforms the other methods with 4.042-order convergence. PWLD and CB-STEP demonstrate convergence orders of 2.997 and 1.01, respectively. These values are all within 0.1 of the theoretical convergence-orders predicted for these methods.

Table III. Convergence of CB-STEP, PWLD, and UCB Methods to Analytical Values for 1D Slab Problems Without Scattering

| Cells_x | SDM/Dimens. | incident flux (n/cm^2*s) | exiting flux (n/cm^2*s) | Incident Flux Error | Exiting Flux Error |
|---|---|---|---|---|---|
| 1 | CBSTEP_1-3 | 5.1758473229E+10 | 5.3578791019E+09 | 8.9648305354E-01 | 3.5663401973E+05 |
| 2 | CBSTEP_2-3 | 9.3806394899E+10 | 6.1946927418E+08 | 8.1238721020E-01 | 4.1232561380E+04 |
| 4 | CBSTEP_4-3 | 1.5797471315E+11 | 4.9649007720E+07 | 6.8405057371E-01 | 3.3037731222E+03 |
| 8 | CBSTEP_8-3 | 2.4009237740E+11 | 3.9948813710E+06 | 5.1981524520E-01 | 2.6491018003E+02 |
| 16 | CBSTEP_16-3 | 3.2440866131E+11 | 4.8628175100E+05 | 3.5118267738E-01 | 3.1368237238E+01 |
| 32 | CBSTEP_32-3 | 3.9350467376E+11 | 1.1007355776E+05 | 2.1299065249E-01 | 6.3267956771E+00 |
| 64 | CBSTEP_64-3 | 4.4040583705E+11 | 4.4055683907E+04 | 1.1918832589E-01 | 1.9324662615E+00 |
| 128 | CBSTEP_128-3 | 4.6831465703E+11 | 2.6321641368E+04 | 6.3370685931E-02 | 7.5204010951E-01 |
| 256 | CBSTEP_256-3 | 4.8363892215E+11 | 2.0008062288E+04 | 3.2722155691E-02 | 3.3179109741E-01 |
| 512 | CBSTEP_512-3 | 4.9168339241E+11 | 1.7365123852E+04 | 1.6633215174E-02 | 1.5586991978E-01 |
| 1024 | CBSTEP_1024-3 | 4.9580682320E+11 | 1.6158438434E+04 | 8.3863535989E-03 | 7.5549653140E-02 |
| 2048 | CBSTEP_2048-3 | 4.9789458322E+11 | 1.5582192604E+04 | 4.2108335691E-03 | 3.7193161876E-02 |
| 4096 | CBSTEP_4096-3 | 4.9894507054E+11 | 1.5300650730E+04 | 2.1098589169E-03 | 1.8452968263E-02 |
| 8192 | CBSTEP_8192-3 | 4.9947197825E+11 | 1.5161501318E+04 | 1.0560435099E-03 | 9.1908045882E-03 |
| 16384 | CBSTEP_16384-3 | 4.9973584965E+11 | 1.5092329052E+04 | 5.2830070920E-04 | 4.5865102466E-03 |
| 1 | PWLD_1-3 | 1.0030060837E+11 | -3.8159322438E+10 | 7.9939878325E-01 | 2.5399893893E+06 |
| 2 | PWLD_2-3 | 1.7572059423E+11 | 4.7915722782E+09 | 6.4855881154E-01 | 3.1893909579E+05 |
| 4 | PWLD_4-3 | 2.7715981031E+11 | 7.9937658118E+06 | 4.4568037938E-01 | 5.3108681528E+02 |
| 8 | PWLD_8-3 | 3.7886190845E+11 | 1.5395473307E+03 | 2.4227618311E-01 | 8.9752353829E-01 |
| 16 | PWLD_16-3 | 4.4905778150E+11 | 1.1660731356E+04 | 1.0188443699E-01 | 2.2382997488E-01 |
| 32 | PWLD_32-3 | 4.8268103378E+11 | 1.4523833607E+04 | 3.4637932443E-02 | 3.3254094318E-02 |
| 64 | PWLD_64-3 | 4.9488230147E+11 | 1.4956538254E+04 | 1.0235397050E-02 | 4.4520949433E-03 |
| 128 | PWLD_128-3 | 4.9860428893E+11 | 1.5014776938E+04 | 2.7914221423E-03 | 5.7556959004E-04 |
| 256 | PWLD_256-3 | 4.9963526031E+11 | 1.5022324302E+04 | 7.2947937981E-04 | 7.3196519048E-05 |
| 512 | PWLD_512-3 | 4.9990675332E+11 | 1.5023285293E+04 | 1.8649335543E-04 | 9.2303390754E-06 |
| 1024 | PWLD_1024-3 | 4.9997642511E+11 | 1.5023406553E+04 | 4.7149788962E-05 | 1.1589293896E-06 |
| 2048 | PWLD_2048-3 | 4.9999407302E+11 | 1.5023421783E+04 | 1.1853952852E-05 | 1.4518312661E-07 |
| 4096 | PWLD_4096-3 | 4.9999851408E+11 | 1.5023423691E+04 | 2.9718454720E-06 | 1.8161276951E-08 |
| 8192 | PWLD_8192-3 | 4.9999962800E+11 | 1.5023423930E+04 | 7.4400879004E-07 | 2.2651603996E-09 |
| 16384 | PWLD_16384-3 | 4.9999990693E+11 | 1.5023423960E+04 | 1.8613329797E-07 | 2.7719136647E-10 |
| 1 | UCB_1-3 | 5.4661462045E+10 | 2.7553988235E+09 | 8.9067707591E-01 | 1.8340584720E+05 |
| 2 | UCB_2-3 | 1.0469267158E+11 | 1.5331432121E+08 | 7.9061465684E-01 | 1.0204018615E+04 |
| 4 | UCB_4-3 | 1.9393325068E+11 | 3.2852743322E+06 | 6.1213349865E-01 | 2.1767680364E+02 |
| 8 | UCB_8-3 | 3.2369101260E+11 | 9.0739202733E+04 | 3.5261797480E-01 | 5.0398483694E+00 |
| 16 | UCB_16-3 | 4.3260793151E+11 | 2.0111341270E+04 | 1.3478413699E-01 | 3.3866562759E-01 |
| 32 | UCB_32-3 | 4.8000314703E+11 | 1.5464006279E+04 | 3.9993705938E-02 | 2.9326358325E-02 |
| 64 | UCB_64-3 | 4.9454622676E+11 | 1.5057396802E+04 | 1.0907546488E-02 | 2.2613245434E-03 |
| 128 | UCB_128-3 | 4.9856479453E+11 | 1.5025795120E+04 | 2.8704109321E-03 | 1.5783056439E-04 |
| 256 | UCB_256-3 | 4.9963059504E+11 | 1.5023580700E+04 | 7.3880991010E-04 | 1.0432726036E-05 |
| 512 | UCB_512-3 | 4.9990619135E+11 | 1.5023434040E+04 | 1.8761729987E-04 | 6.7069109281E-07 |
| 1024 | UCB_1024-3 | 4.9997635633E+11 | 1.5023424603E+04 | 4.7287347810E-05 | 4.2518379177E-08 |
| 2048 | UCB_2048-3 | 4.9999406452E+11 | 1.5023424005E+04 | 1.1870955008E-05 | 2.6973701366E-09 |
| 4096 | UCB_4096-3 | 4.9999851302E+11 | 1.5023423966E+04 | 2.9739584020E-06 | 1.1817782188E-10 |
| 8192 | UCB_8192-3 | 4.9999962786E+11 | 1.5023423949E+04 | 7.4427212598E-07 | 1.0433550096E-09 |
| 16384 | UCB_16384-3 | 4.9999990692E+11 | 1.5023424032E+04 | 1.8616616602E-07 | 4.5278250414E-09 |
| analytical | | 5.0000000000E+11 | 1.5023423964E+04 | | |

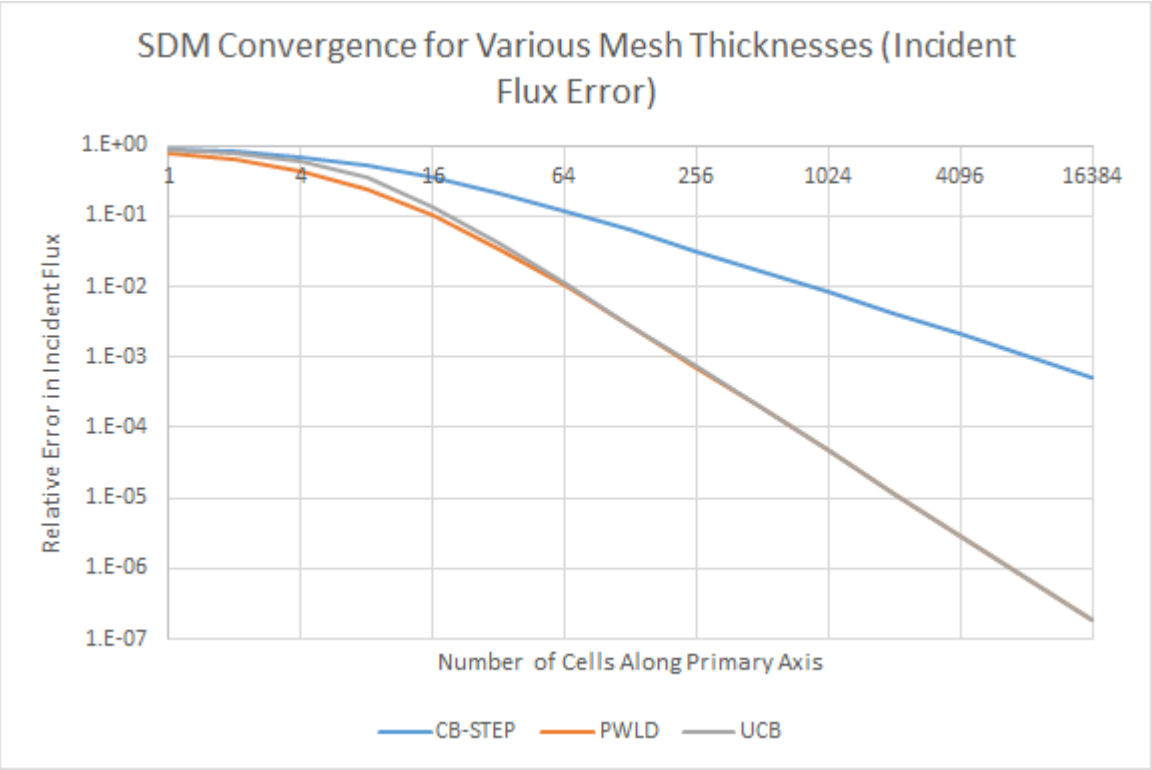



Fig. III.10. $\log_2$-$\log_{10}$ plot of the numerical error of CB-STEP, UCB, and PWLD spatial discretization methods for the incident flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with no scattering.

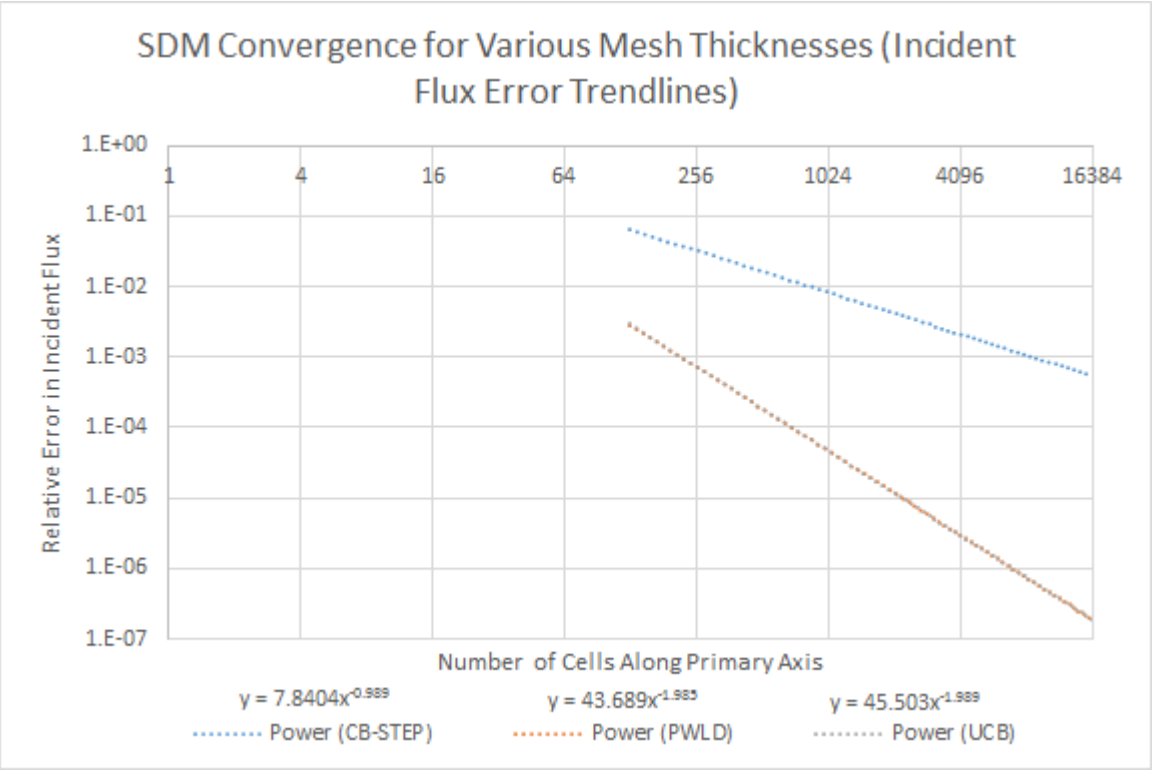



Fig. III.11. $\log_2$-$\log_{10}$ plot of the numerical error trendlines of CB-STEP, UCB, and PWLD spatial discretization methods for the incident flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with no scattering.

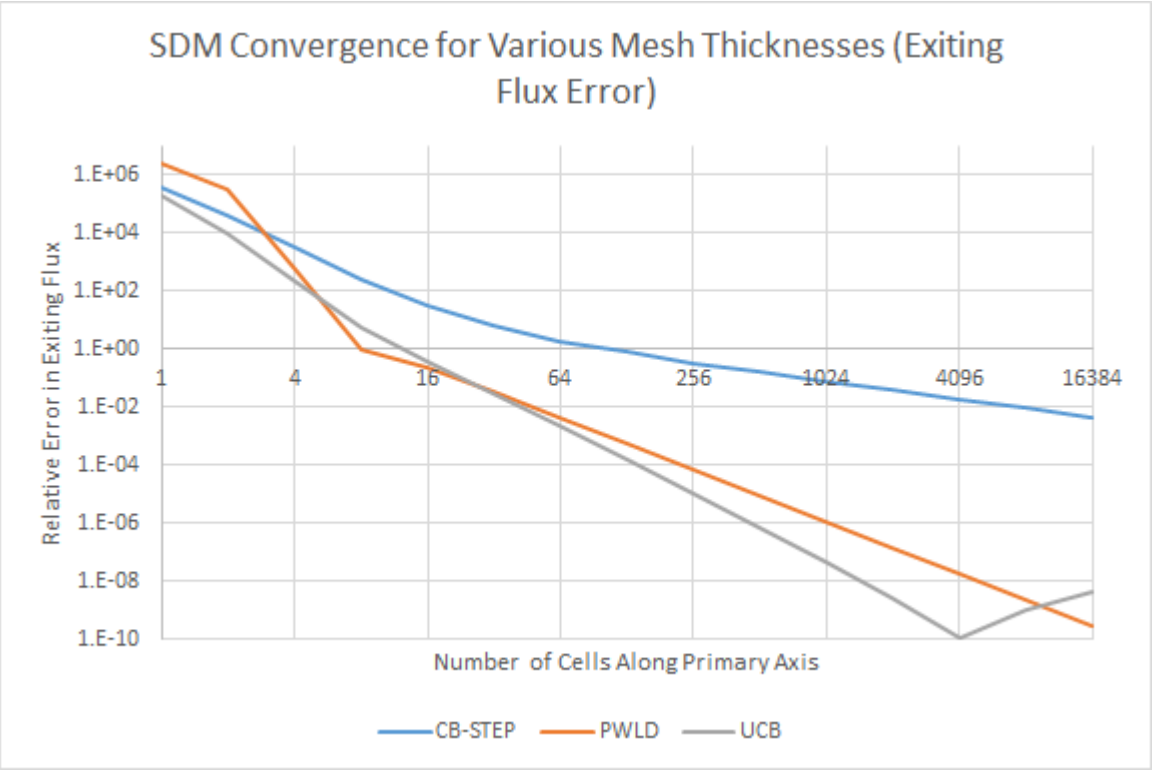



Fig. III.12. $\log_2$-$\log_{10}$ plot of the numerical error of CB-STEP, UCB, and PWLD spatial discretization methods for the exiting flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with no scattering.

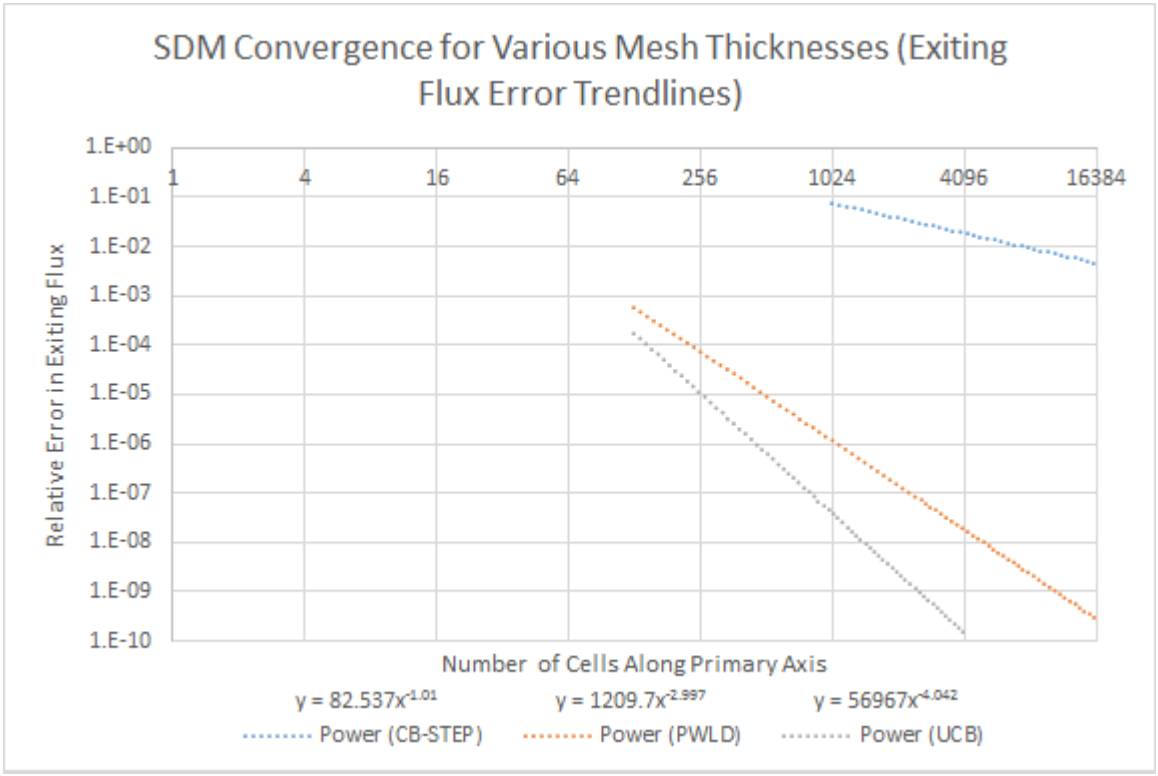



Fig. III.13. $\log_2$-$\log_{10}$ plot of the numerical error trendlines of CB-STEP, UCB, and PWLD spatial discretization methods for the exiting flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with no scattering.

Table IV. Comparison of Convergence Orders of CB-STEP, PWLD, and UCB Methods for Non-Scattering 1D Slab Problems Without Scattering

| Method | Incident Flux Error Convergence Order | Exiting Flux Error Convergence Order |
|---|---|---|
| CB-STEP | 0.989 | 1.01 |
| PWLD | 1.985 | 2.997 |
| UCB | 1.989 | 4.042 |

Moving on to the scattering problem, Table V below shows the outputs of each of the three methods and the numerical error observed for each.

Figure III.14 shows the numerical error estimate associated with each method's values for the incident flux as a function of the number of cells in the direction of interest.

Figure III.15 shows the trendlines derived from the data shown in Figure III.12 and their equations, which give the order of convergence for each method.

Figure III.16 shows the numerical error estimate associated with each method's values for the exiting flux as a function of the number of cells in the direction of interest.

Figure III.17 shows the trendlines derived from the data shown in Figure III.14 and their equations, which give the order of convergence for each method.

When trendlines are calculated for the this data, the orders of convergence for each method can be determined. Table VI lists the convergence orders of the incident and exiting fluxes for each method. In addition, the convergence orders generated for the scattering case are only useful for comparisons, as the relative errors are calculated using an averaged pseudo-analytical value. When calculating the incident flux in the scattering case, UCB and PWLD share approximately 2nd-order convergence, although UCB converges slightly faster. CB-STEP demonstrates single-order convergence. However, important to note again is the fact that the incident flux is user-defined and therefore already known with no error, so the convergence order of the exiting flux is the more important figure for characterizing each method. When scattering is taken into account, the convergence orders of these methods change from those observed in the non-scattering case. For 1D slab problems without scattering, UCB still outperforms the other methods, but now demonstrates with 3.48-order convergence. The convergence orders of PWLD and CB-STEP have also decreased to 2.971 and 1.047, respectively. Important to note is that although the convergence orders of all three methods decrease when scattering is taken into account, their relative convergences do not. UCB still outperforms the other methods although its order of convergence by far decreases by the most from the non-scattering case.

Table V. Convergence of CB-STEP, PWLD, and UCB Methods to Analytical Values for 1D Slab Problems with Scattering

| Cells_x | SDM/Dimens. | incident flux (n/cm^2*s) | exiting flux (n/cm^2*s) | Incident Flux Error | Exiting Flux Error |
|---|---|---|---|---|---|
| 1 | CBSTEP_1-3 | 5.1858638817E+10 | 4.0939730289E+09 | 9.1145131351E-01 | 3.6656481789E+08 |
| 2 | CBSTEP_2-3 | 9.4394063917E+10 | 2.9089798950E+08 | 8.3882202535E-01 | 2.6046328100E+07 |
| 4 | CBSTEP_4-3 | 1.6071261720E+11 | 1.0447622508E+07 | 7.2558301799E-01 | 9.3545480919E+05 |
| 8 | CBSTEP_8-3 | 2.4950816385E+11 | 2.7011258459E+05 | 5.7396451813E-01 | 2.4184252310E+04 |
| 16 | CBSTEP_16-3 | 3.4742528808E+11 | 8.9625217740E+03 | 4.0677091386E-01 | 8.0148334509E+02 |
| 32 | CBSTEP_32-3 | 4.3462994471E+11 | 6.2596914166E+02 | 2.5786886058E-01 | 5.5047820400E+01 |
| 64 | CBSTEP_64-3 | 4.9835380043E+11 | 1.0595019383E+02 | 1.4906030234E-01 | 8.4865338241E+00 |
| 128 | CBSTEP_128-3 | 5.3832105840E+11 | 3.6959566207E+01 | 8.0816162577E-02 | 2.3092735584E+00 |
| 256 | CBSTEP_256-3 | 5.6098585479E+11 | 2.0726640144E+01 | 4.2115996192E-02 | 8.5581513044E-01 |
| 512 | CBSTEP_512-3 | 5.7310007292E+11 | 1.5295147223E+01 | 2.1430954565E-02 | 3.6949189264E-01 |
| 1024 | CBSTEP_1024-3 | 5.7936914403E+11 | 1.3087717699E+01 | 1.0726508308E-02 | 1.7184378941E-01 |
| 2048 | CBSTEP_2048-3 | 5.8255892486E+11 | 1.2094250318E+01 | 5.2799538016E-03 | 8.2891031794E-02 |
| 4096 | CBSTEP_4096-3 | 5.8416792156E+11 | 1.1623169054E+01 | 2.5325900458E-03 | 4.0711511552E-02 |
| 8192 | CBSTEP_8192-3 | 5.8497598422E+11 | 1.1393814251E+01 | 1.1528221165E-03 | 2.0175616144E-02 |
| 16384 | CBSTEP_16384-3 | 5.8538090787E+11 | 1.1168463533E+01 | 4.6141449948E-04 | 1.7627408138E-06 |
| 1 | PWLD_1-3 | 1.0078930988E+11 | -4.2110612327E+10 | 8.2790213538E-01 | 3.7704862431E+09 |
| 2 | PWLD_2-3 | 1.7889723461E+11 | 5.7351765651E+09 | 6.9453276247E-01 | 5.1351436367E+08 |
| 4 | PWLD_4-3 | 2.8922554490E+11 | 5.3653827439E+07 | 5.0614704349E-01 | 4.8040378638E+06 |
| 8 | PWLD_8-3 | 4.0989635007E+11 | 4.2594554810E-06 | 3.0010150240E-01 | 9.9999961862E-01 |
| 16 | PWLD_16-3 | 5.0388784054E+11 | 3.6867863103E+00 | 1.3961092237E-01 | 6.6989373243E-01 |
| 32 | PWLD_32-3 | 5.5537481191E+11 | 9.6739764271E+00 | 5.1696858480E-02 | 1.3381466074E-01 |
| 64 | PWLD_64-3 | 5.7625266229E+11 | 1.0955321658E+01 | 1.6047904524E-02 | 1.9085990315E-02 |
| 128 | PWLD_128-3 | 5.8309646562E+11 | 1.1140354658E+01 | 4.3621023275E-03 | 2.5185659623E-03 |
| 256 | PWLD_256-3 | 5.8507060451E+11 | 1.1164885385E+01 | 9.9125785975E-04 | 3.2214178524E-04 |
| 512 | PWLD_512-3 | 5.8560171387E+11 | 1.1168041103E+01 | 8.4387996660E-05 | 3.9586155092E-05 |
| 1024 | PWLD_1024-3 | 5.8573951314E+11 | 1.1168441358E+01 | 1.5090441146E-04 | 3.7482006263E-06 |
| 2048 | PWLD_2048-3 | 5.8577461210E+11 | 1.1168491762E+01 | 2.1083592254E-04 | 7.6477349089E-07 |
| 4096 | PWLD_4096-3 | 5.8578346935E+11 | 1.1168498085E+01 | 2.2595969422E-04 | 1.3310011133E-06 |
| 8192 | PWLD_8192-3 | 5.8578569407E+11 | 1.1168498877E+01 | 2.2975839717E-04 | 1.4019113868E-06 |
| 16384 | PWLD_16384-3 | 5.8578624805E+11 | 1.1168487158E+01 | 2.3070432215E-04 | 3.5259698400E-07 |
| 1 | UCB_1-3 | 5.4500084834E+10 | 1.5292244635E+09 | 9.0694104134E-01 | 1.3692319917E+08 |
| 2 | UCB_2-3 | 1.0336710939E+11 | 5.5928307090E+07 | 8.2350053970E-01 | 5.0076894793E+06 |
| 4 | UCB_4-3 | 1.8773915446E+11 | 6.2569777218E+05 | 6.7943517397E-01 | 5.6022522608E+04 |
| 8 | UCB_8-3 | 3.1564378738E+11 | 3.3824814899E+03 | 4.6103786353E-01 | 3.0185952203E+02 |
| 16 | UCB_16-3 | 4.5747557937E+11 | 6.0852808056E+01 | 2.1885991266E-01 | 4.4486188372E+00 |
| 32 | UCB_32-3 | 5.4457316811E+11 | 1.4495262779E+01 | 7.0140677927E-02 | 2.9787210075E-01 |
| 64 | UCB_64-3 | 5.7472231180E+11 | 1.1469324704E+01 | 1.8660979777E-02 | 2.6936646459E-02 |
| 128 | UCB_128-3 | 5.8291705146E+11 | 1.1194446455E+01 | 4.6684522166E-03 | 2.3246876732E-03 |
| 256 | UCB_256-3 | 5.8505020788E+11 | 1.1170737469E+01 | 1.0260851315E-03 | 2.0184022632E-04 |
| 512 | UCB_512-3 | 5.8559934346E+11 | 1.1168705052E+01 | 8.8435485619E-05 | 1.9862347145E-05 |
| 1024 | UCB_1024-3 | 5.8573922989E+11 | 1.1168519760E+01 | 1.5042077141E-04 | 3.2716859111E-06 |
| 2048 | UCB_2048-3 | 5.8577457757E+11 | 1.1168501263E+01 | 2.1077696712E-04 | 1.6155078756E-06 |
| 4096 | UCB_4096-3 | 5.8578346509E+11 | 1.1168499254E+01 | 2.2595242195E-04 | 1.4356307552E-06 |
| 8192 | UCB_8192-3 | 5.8578569354E+11 | 1.1168499021E+01 | 2.2975749343E-04 | 1.4147474003E-06 |
| 16384 | UCB_16384-3 | 5.8578625148E+11 | 1.1168498969E+01 | 2.3071017733E-04 | 1.4101438297E-06 |
| Average | | 5.8565113580E+11 | 1.1168483220E+01 | | |

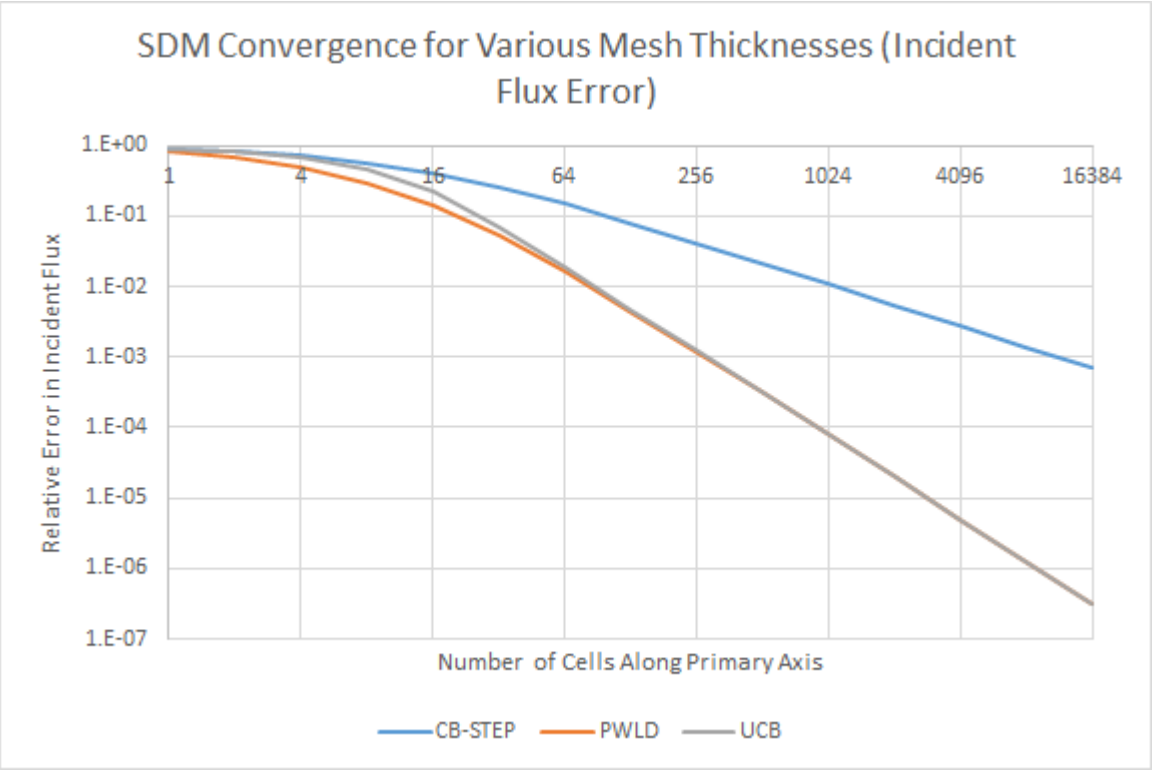



Fig. III.14. $\log_2$-$\log_{10}$ plot of the numerical error of CB-STEP, UCB, and PWLD spatial discretization methods for the incident flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with scattering.

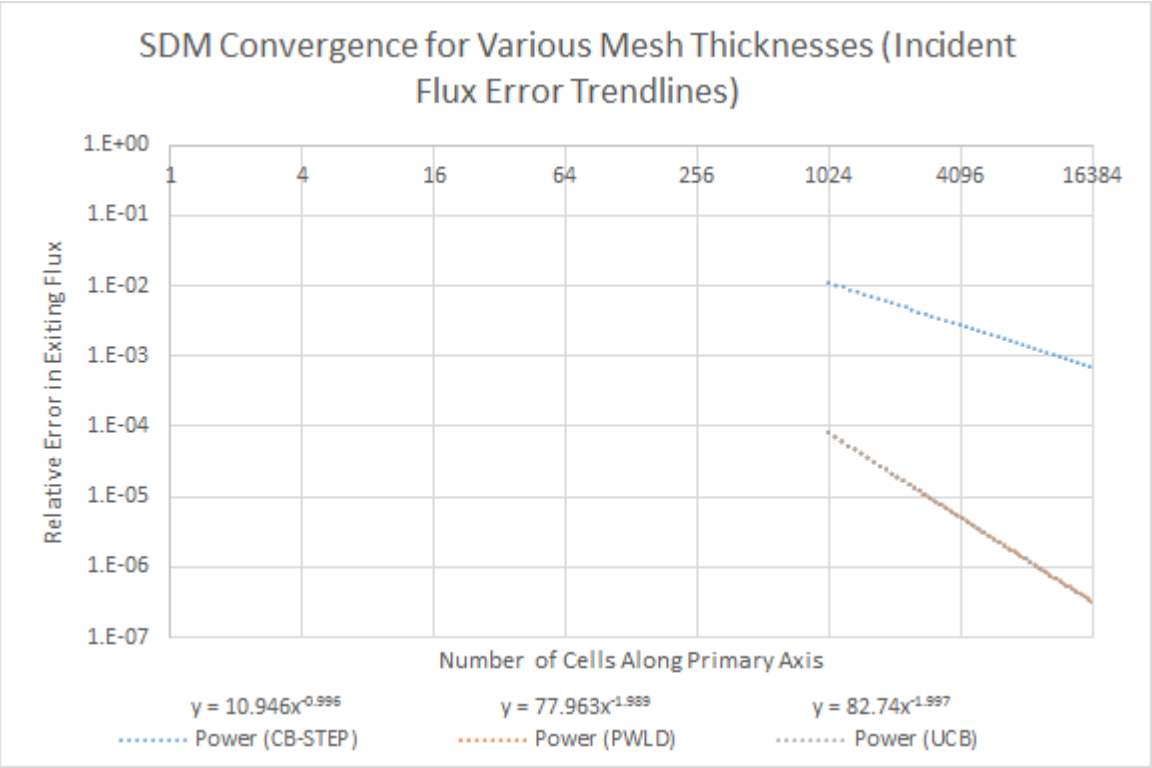



Fig. III.15. $\log_2$-$\log_{10}$ plot of the numerical error trendlines of CB-STEP, UCB, and PWLD spatial discretization methods for the incident flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with scattering.

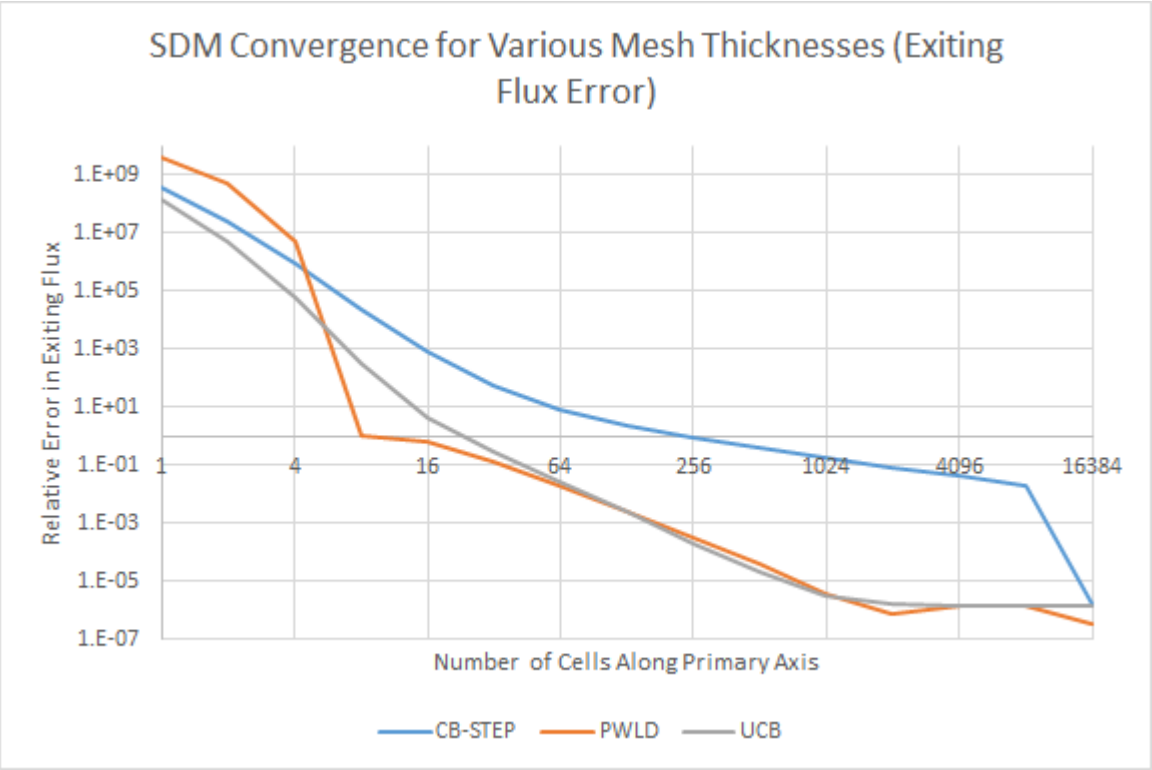



Fig. III.16. $\log_2$-$\log_{10}$ plot of the numerical error of CB-STEP, UCB, and PWLD spatial discretization methods for the exiting flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with scattering.

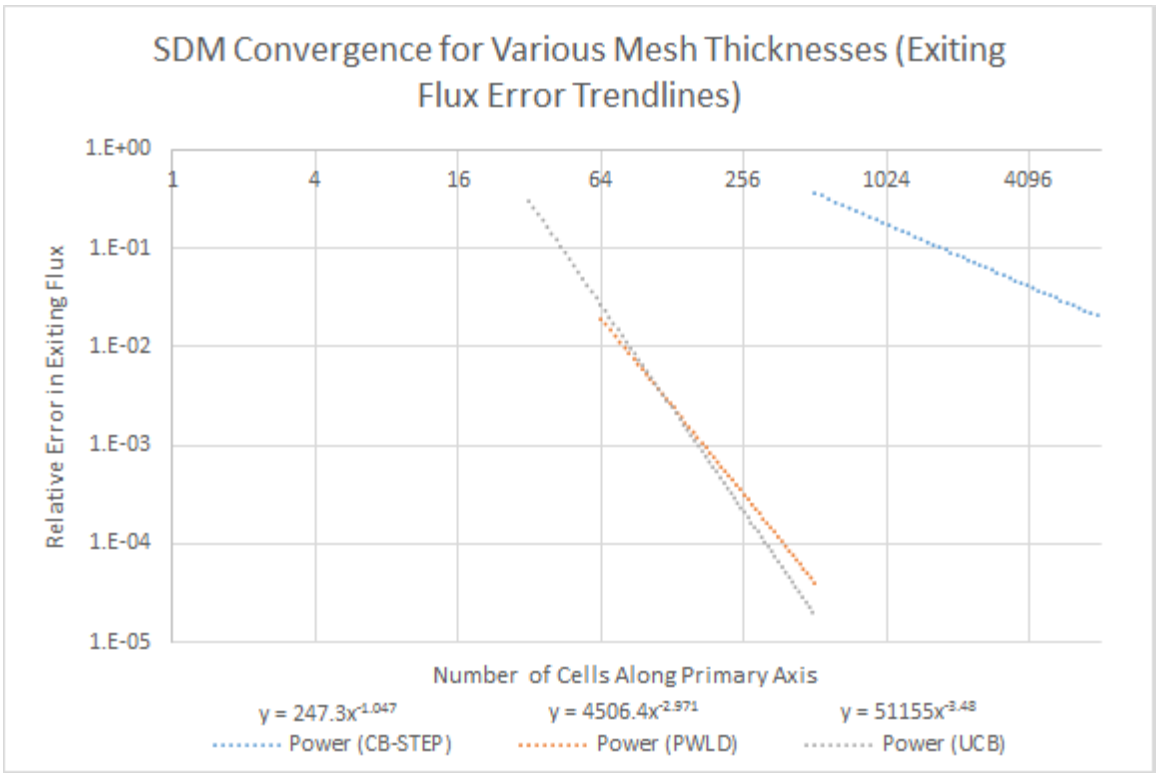



Fig. III.17. $\log_2$-$\log_{10}$ plot of the numerical error trendlines of CB-STEP, UCB, and PWLD spatial discretization methods for the exiting flux as a function of the number of cells in the direction of interest (mesh fineness) for the 1D slab problem with scattering.

Table VI. Comparison of Convergence Orders of CB-STEP, PWLD, and UCB Methods for 1D Slab Problems with Scattering

| Method | Incident Flux Error Convergence Order | Exiting Flux Error Convergence Order |
|---|---|---|
| CB-STEP | 0.996 | 1.047 |
| PWLD | 1.989 | 2.971 |
| UCB | 1.997 | 3.48 |

# CHAPTER IV

# CONCLUSION

This project sought to provide a qualitative discussion of the spatial discretization methods currently in use for particle transport simulations, and to detail the implementation and testing of the upstream corner balance (UCB) family of methods, in particular the derivative suggested in 1997 by Adams. [7] This method was tested against other methods currently in use in an effort to determine whether it represents an improvement over such methods and therefore merits future use on a larger scale.

The implementation phase of this project proved to be the most time-consuming, and unfortunately limited the breadth and depth of the other endeavors of this research. However, the method was successfully implemented and, to lay the foundation for the continuation of this research, intentionally implemented in an extremely plastic way. Looking at this work qualitatively, we are optimistic that a derivative of this method not yet tested may result in a significant improvement over current methods, and therefore, have provided for the straightforward incorporation of new UCB derivatives into the method family should they be developed.

After testing, we believe that this implementation of UCB has been acceptably verified. The small number of sufficiently simple problems with known solutions limits the extent to which these processes can continue, so once the mathematics of a method have been sufficiently proven physical and the implementation of that method sufficiently proven correct, other simulations must commence, although it is always wise to corroborate one method with another.

Once UCB was confidently verified, it was compared to two other currently employed methods, the piecewise linear discontinuous (PWLD) method and the corner balance-step (CB-STEP) method. All three methods were used to simulate a one-dimensional slab problem with and without scattering, and by altering the fineness of the mesh employed, each prob-

lem's order of convergence to the analytical solution was quantified. From this testing process, it was determined that UCB greatly outperformed CB-STEP in all test cases, and outperformed PWLD in all test cases as well (although by a smaller margin). Based on this limited testing, we believe UCB holds promise for both scattering and non-scattering problems, and should continue to be tested against other methods for non-scattering problems.

Therefore, at the conclusion of this research, the data suggest a few important recommendations. First, UCB holds promise as a method, and should be tested on other problems of varying complexities to determine areas in which it may outperform current method. One possible continuation of this work involves the massively parallel simulation of pressurized water reactor (PWR) core assemblies using UCB in order to facilitate a more in-depth comparison with other currently employed methods. In addition, although analytical solutions do not exist for problems of such complexity, accepted solutions based on very high-fidelity simulations by many research groups using a variety of methods provide another opportunity for assessing UCB performance on problems of practical interest. Second, the plasticity incorporated into this implementation of the UCB family of methods leaves ample opportunity for the streamlined implementation of other UCB variants, any of which holds the potential for improvement over currently employed methods. Thus, a second possible continuation of this work involves the development, implementation, and comparative testing of other UCB methods in a manner similar to this project.

Unfortunately, we must acknowledge that time constraints and unforeseen impediments curtailed the original aspirations of this project, and at this time, sufficiently data does not exist to recommend the imminent implementation of UCB on a large scale within the field of particle transport simulation. However, as mentioned, the data is sufficiently encouraging to warrant the strongest recommendation for future testing with the hopes that a robust, accurate, and efficient spatial discretization method will result.

# Bibliography


[1] M. L. Adams, "Even-parity finite-element transport methods in the diffusion limit," *Progress in Nuclear Energy*, vol. 25, no. 2-3.

[2] M. L. Adams, "Discontinuous finite-element transport solutions in the thick diffusion limit in cartesian geometry," vol. 5, 1991.

[3] T. M. Pandya*, M. L. Adams, and W. D. Hawkins, "Long characteristics with piecewise linear sources designed for unstructured grids," *Proc. International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering, Rio de Janeiro, Brazil*, 2011.

[4] M. L. Adams, T. A. Wareing, and W. F. Walters, "Characteristic methods in thick diffusive problems," *Nuclear Science and Engineering*, vol. 130, pp. 18–46, 1998.

[5] E. W. Larsen and J. E. Morel, "Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii," *Journal of Computational Physics*, vol. 83, p. 212, 1989.

[6] E. W. Larsen, J. E. Morel, and W. F. Miller Jr., "Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes," *Journal of Computational Physics*, vol. 69, p. 283, 1987.

[7] M. L. Adams, "Subcell balance methods for radiative transfer on arbitrary spatial grids," *Transport Theory and Stat. Phys*, vol. 26, no. 4-5.

[8] W. D. Hawkins, T. Smith, M. P. Adams, L. Rauchwerger, N. Amato, and M. L. Adams, "Provably optimal parallel transport sweeps on regular grids," *Trans. Amer. Nucl. Soc.*, vol. 107, no. 1.

[9] M. P. Adams, M. L. Adams, W. D. Hawkins, T. Smith, L. Rauchwerger, N. M. Amato, T. S. Bailey, and R. D. Falgout, "Efficient massively parallel transport sweeps," *Proc.*

*International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering, Sun Valley, Idaho, USA, May 5-9, CD-ROM*, 2013.